



Software Engineering Institute

Requirements and Their Impact Downstream: Improving Causal Analysis Processes Through Measurement and Analysis of Textual Information

Ira A. Monarch
Dennis R. Goldenson
Lawrence T. Osiecki

September 2008

TECHNICAL REPORT
CMU/SEI-2008-TR-018
ESC-TR-2008-018

Software Engineering Measurement and Analysis Initiative

Unlimited distribution subject to the copyright.

<http://www.sei.cmu.edu>



CarnegieMellon

This report was prepared for the

SEI Administrative Agent
ESC/XPB
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2008 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be directed to permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about SEI reports, please visit the publications section of our website (<http://www.sei.cmu.edu/publications>).

Table of Contents

Acknowledgments	v
Abstract	vii
1 Introduction and Executive Summary	1
1.1 Overview and Interpretation of the Results	2
1.2 Audience	3
1.3 This Document	3
2 Requirements-Related Problems and Impacts	5
2.1 Capabilities and Quality Attributes	5
2.2 Requirements Engineering: Multiple Stakeholders	6
3 Semi-Automated Content Analysis: Applying the Method	9
4 Analysis and Results	13
4.1 Requirements, Validation, and Usability at the SEC	13
4.2 An Initial Semantic Categorization	14
4.3 Quality Attributes in the Three Projects	15
4.4 Project 1	16
4.5 Project 2	25
4.6 Project 3	33
4.7 Synopsis and Future Opportunities	40
5 Summary and Conclusions	43
5.1 Implications and Limitations	44
5.2 Conclusions and Future Work	45
Appendix A: Further Background on Quality Attributes and Usability	49
Appendix B: Further Background on Semi-Automated Content Analysis	53
References/Bibliography	57

List of Figures

Figure 1:	Stakeholder Perspective Differences in Commercial Software	7
Figure 2:	Example Concept Map Showing Themes	10
Figure 3:	Example Concept Map Showing Both Concepts & Themes	11
Figure 4:	Semantic Categories	15
Figure 5:	Semantic Categories in Project 1	16
Figure 6:	Project 1 PCRs Concept Map Showing Themes Only	18
Figure 7:	Project 1 PCRs Concept Map Showing Both Concepts & Themes	19
Figure 8:	Links Among Key Concepts Derived from Project 1 PCRs	20
Figure 9:	Project 1 PCR's Ranked Concepts and Concepts Related to Screen	21
Figure 10:	Many PCRs Point to Issues Providing Feedback with Respect to Data Entry Errors	22
Figure 11:	Ordering of Buttons on the Mission Menu	24
Figure 12:	Semantic Categories in Project 2	25
Figure 13:	Project 2 PCRs Concept Map Showing Themes Only	26
Figure 14:	Project 2 PCRs Concept Map Showing Both Themes and Concepts	28
Figure 15:	Q_C_A_Usability Concept Links	29
Figure 16:	Ranked Concept List & Text Block Counts for Concepts Co-Occurring with Q_C_A Usability	30
Figure 17:	Less Frequent Concepts Collected in the Q_C_A_Usability Theme – Focus on stack_dump	32
Figure 18:	Browsing a PCR with the Co-occurring Concepts stack-dump & Q_C_A_Usability	33
Figure 19:	Semantic Categories in Project 3	34
Figure 20:	Project 3 PCRs Concept Map Showing Themes Only	35
Figure 21:	Project 3 PCRs Concept Map Showing Themes and Concepts	36
Figure 22:	Most Frequent Concepts Populating the Theme DRB in Project 3 PCRs	37
Figure 23:	Most Frequent Concepts in all Themes Interrelating with DRB in Project 3 PCRs	38
Figure 24:	Ranked Concept List for Project 3 PCRs	39
Figure 25:	Displays, Shows, Mission and Data Themes in Project 3 PCRs	40
Figure 26:	ISO/IEC 9126-1: Software Product Quality	50
Figure 27:	SEI Software Architecture Quality Attribute Scenarios	51
Figure 28:	Understanding a Key Quality Attribute – Krippendorff's Usability	52

Acknowledgments

Many thanks to the dedicated people of the US Army RDECOM-ARDEC Armament Software Engineering Center who have taken time out of their busy schedules for this work, in particular Nathan Becker and Julie Robbins. Special thanks also are due to our SEI colleagues Bob Ferguson, Erin Harper, Mike Konrad, Jim McCurley, Mike Phillips, Bob Stoddard, and Dave Zubrow.

Abstract

Requirements documents, test procedures, and problem and change reports from a U. S. Army Software Engineering Center (SEC) were analyzed to identify, clarify, and begin categorizing recurring patterns of issues raised throughout the product life cycle. Semi-automated content analysis was used to identify underlying patterns in the SEC documents. Automated tools and techniques were used to support efficient search and related semantic analysis that would not be possible manually. Discussions with Army personnel were used to confirm and elaborate initial findings and interpretations. The same analytic methods can be used as a basis for novel, proactive causal analysis processes.

One of the patterns identified suggests that usability is not sufficiently articulated and quantified early in the product life cycle. While the SEC has established exemplary processes to handle usability-related issues when they arise, some of them might be mitigated or prevented by documented consideration upstream.

1 Introduction and Executive Summary

Problems associated with requirements development, analysis, management, and validation have been pervasive in software and systems engineering for many years. Several recent reviews—like many before them—have identified requirements as among the top challenges facing software and systems engineering today [USGAO 2008, USGAO 2004, Walker 2007]. Yet there are many reasons why requirements engineering continues to be so difficult. Product requirements are typically emphasized instead of customer or user operational needs, and effective processes and infrastructure for tracing developing capabilities and requirements across the product life cycle tend to be deficient.¹ The problem may be due to differing perspectives and priorities among key stakeholders, including users, acquirers, maintainers, and developers.

Problem reports and change requests can be rich sources of data for investigating and improving the adequacy of characterizations of usability and other quality attributes. Yet researchers typically do not have access to such data, and practitioners often lack the resources to trace them back to requirements, analyze them for lessons learned, or recognize recurring problem areas.

The work described in this report is based largely on collaboration with one Army Software Engineering Center (SEC). The collaboration enabled researchers to access the necessary data and increased practitioner awareness that content analysis of textual documentation might be worth pursuing. The work uses semi-automated content analysis methods and tools to analyze requirements documents, testing procedures, and problem and change reports (PCRs). The tools capture recurring themes in the text that might be missed by manual methods alone. Interpretations of the content analysis results are then corroborated and refined through interviews with the domain experts and stakeholders who produce and use the documentation.

The overall aim of the report is to provide

1. an improved understanding of requirements and requirements-related issues in testing and maintenance
2. help in judging the potential of semi-automated content analysis to enable increased understanding and improvement of requirements engineering and other difficult aspects of software and systems engineering

Difficulties in validating requirements and tracing their effects downstream are pervasive in software and systems engineering. As will be seen later, the SEC has established exemplary processes to handle such problems when they arise. The initial semi-automated content analyses described in this report demonstrate its potential to efficiently identify recurring themes that might otherwise

¹ Two technical reports are in progress that address this work: *Capabilities Engineering Framework: Elaboration Through Case Studies* by Ira Monarch and *Capabilities Engineering Framework: A Holistic Guide to Quality-Driven System of System Life-Cycle Engineering* by Ira Monarch and James Wessel.

go unnoticed under the demands of day-to-day work. Future work with the SEC will examine the use of semi-automated content analyses in a novel approach to proactive causal analysis.²

This report introduces the workings of semi-automated content analysis methods and tools. It shows how they can be used to provide practical value in support of engineering and management decisions by deriving useful quantitative information from qualitative textual sources. Methods that combine qualitative and quantitative analyses to help identify and define measurable concepts may become the basis for a useful and credible approach to empirical software and systems engineering.

1.1 OVERVIEW AND INTERPRETATION OF THE RESULTS

The examples described in this report come from three Army SEC projects. The results of the automated text analyses using documentation provided by these projects cohere well with the observations reported in interviews and other discussions with the projects' subject matter experts who have been collaborating in this work. The analyses, corroborated by them, also identified issues that are now being addressed along with other issues worthy of being addressed in the future (see Section 5.2).

As expected, the requirements specifications, test procedures, and PCRs contained some explicit quality-attribute-related terminology. Because the systems maintained by these three projects are used in the heat of battle, both security and accuracy are discussed in the requirement specifications. "Usability" and "readability" also were explicitly identified in one project's PCRs for a period of time.

Issues related to usability (e.g., screen, display, button press, and menu) are identified as recurring concepts by the initial text analyses in all three projects and all three types of documents analyzed. While the requirements documents do not capture usability as a quality attribute per se, issues involving usability are sometimes explicit and quite often implied in all three sets of PCRs. There are also PCR issues related to usability that were not explicitly addressed in the testing procedures. Moreover, usability appears as a major theme in the PCR content analysis because of its pervasive association with several other concepts and themes identified through the analysis. Other issues related to quality attributes, such as maintainability and reusability, often are described in the same PCRs.

Clearly stated usability criteria for information technology typically are not considered sufficiently early in, or indeed throughout, the system life cycle. There are several reasons why that is so and why such criteria are difficult to measure and apply: (1) functional criteria are better understood and expressed in the requirements, (2) the functionality is perceived to be so important that users have an immediate need for it and will learn to use it, or (3) developers believe that each usability related issue can be fixed as it arises or deferred if it is not urgent.

This problem persists for methodological as well as substantive reasons. It is difficult to conceptualize usability, much less measure it quantitatively. While general guidelines have been speci-

² The method can be employed to support the CMMI Organizational Innovation and Deployment process area as well as Causal Analysis and Resolution. It also may be incorporated into the SEC's ongoing Six Sigma work.

fied for defining usability and other quality attributes, conceptual and operational definitions also must be sensitive to the specific contexts where the definitions are used. Actual cases need to be studied to determine the adequacy of existing definitions related to these and other aspects of requirements engineering.

Much more needs to be known about the prevalence and significance of usability issues in the SEC as well as elsewhere in software and systems engineering. The same is so about the extent to which process improvements and the inclusion of usability acceptance criteria in requirements specifications would serve as a basis for avoiding or mitigating usability-related problems.

1.2 AUDIENCE

Both practitioners and researchers of systems and software engineering may benefit from reading this report. In fact, it is our hope to blur the distinction between the two groups. The analytic approach and methods described here are meant to be adopted by practitioners, and the analyses are based on the terminology that practitioners use in their own context of work. Similarly, by moving the focus of research from high-level abstractions and generalization to actual context of use, the results of software and systems engineering research may become more useful for practitioners.

The report is meant for practitioners involved in requirements traceability and validation as a part of their capability-driven maintenance and testing activities. In the U. S. Army, these include, in particular, material developers associated with the SECs.

Researchers involved in software, systems, and requirements engineering research are the other primary audience for whom this report is meant. The document will introduce them to the fundamentals for doing research using semi-automated content analysis. It will help them explain results based on such methods to others, thus helping practitioners improve their processes and infrastructure on the basis of recurring patterns as opposed to isolated problem solving. It also will provide the researchers with a basis to further develop their own skills in performing similar research in domain-specific and context-sensitive conceptualization and measurement.

In addition to the typical skills of empirical software engineering researchers, some familiarity with information science and semantic analysis is helpful in evaluating the scientific underpinnings of the content analysis tools, semantic analysis, and ontological techniques. However, experienced researchers will be able to appreciate and understand the document at a conceptual level and will be in a position to delve more deeply into these fields after reading it. References included in the bibliography provide a roadmap to further work in the field.

1.3 THIS DOCUMENT

The remainder of this document includes four sections and two appendices. Section 2 provides brief discussions of problems that are often faced in requirements development and the impact of those requirements downstream. The emphasis is on the importance of capabilities and quality attributes, as well as the existence of multiple stakeholders with different needs and perspectives. A brief introduction to semi-automated content analysis can be found in Section 3. The results of that analysis are described in detail in Section 4. Section 5 contains summaries of the results,

conclusions, and suggestions for future work. Further background about quality attributes and usability is provided in Appendix A. Appendix B contains further background about semi-automated content analysis.

2 Requirements-Related Problems and Impacts

2.1 CAPABILITIES AND QUALITY ATTRIBUTES

Mutual understanding of capabilities and requirements across multiple stakeholders over the entire project life cycle is very difficult to achieve. A well-understood way of establishing and sustaining an evolving common language of concepts, relations, and attributes to characterize the desired operational capabilities could significantly mitigate the problem. Quality attributes thus can play a pivotal role.

Recent emphasis on quality attributes for software and system architectures have made developers, and to a certain extent acquirers, more aware of the potential benefits of considering such “non-functional” requirements earlier in the product life cycle [Ozkaya 2008]. Yet anecdotal evidence and our initial analyses show that this awareness remains limited to certain types of quality attributes. In particular, usability tends not to be considered sufficiently by practitioners. This may contribute to otherwise preventable or reducible downstream problems such as rework, cost and schedule overruns, and especially reduced usefulness or fitness for use of the resulting technology.

Further use of processes to formulate and negotiate quality attributes across stakeholders might help provide needed common ground across customer, contractual, and product requirements [Barbacci 2003]. Quality attribute and similar conceptual frameworks have been developed and used for several purposes. These include frameworks for software architecture as well as industry and international standards [ISO/IEC 1991, Bass 2003b, Firesmith 2006]. Many papers and books exist that focus heavily on various individual attributes such as usability, security, and interoperability [Bass 2003a, Ellison 2004, Krippendorff 2006, O’Brien 2005]. A similar approach to capabilities is an essential part of the Joint Capabilities Integration and Development System (JCIDS) policy [CJCSI 2007]. There also is a voluminous literature and many consultancies on usability and human computer interaction [Shneiderman 2004, Myers 1998, Nielsen 1994].

Further detail follows in Appendix A. However, suffice it to say that there is no single definitive statement of a quality attributes typology, much less a shared underlying ontology [Masolo 2003, Guarino 2005]. This may be particularly true with respect to usability. These are difficult concepts that are not well or widely understood. Army and other Department of Defense (DoD) organizations sometimes struggle in interpreting and using the JCIDS policy on capabilities, and standards groups continue to struggle with categorizing and clearly defining their quality attribute models and terminology.

These concepts are defined in broad, general terms that can be difficult to translate into terminology that is meaningful and precise enough to be useful in practical circumstances. In addition, there can be important tradeoffs among quality attributes; they do not exist in isolation (e.g., usability *versus* reusability or maintainability or the immediate need for new functionality to support the warfighter) [Bass 2003b].

Such issues are not always fully considered in more general discussions of quality attributes. The incorporation of semi-automated content analysis in a new, proactive approach to causal analysis

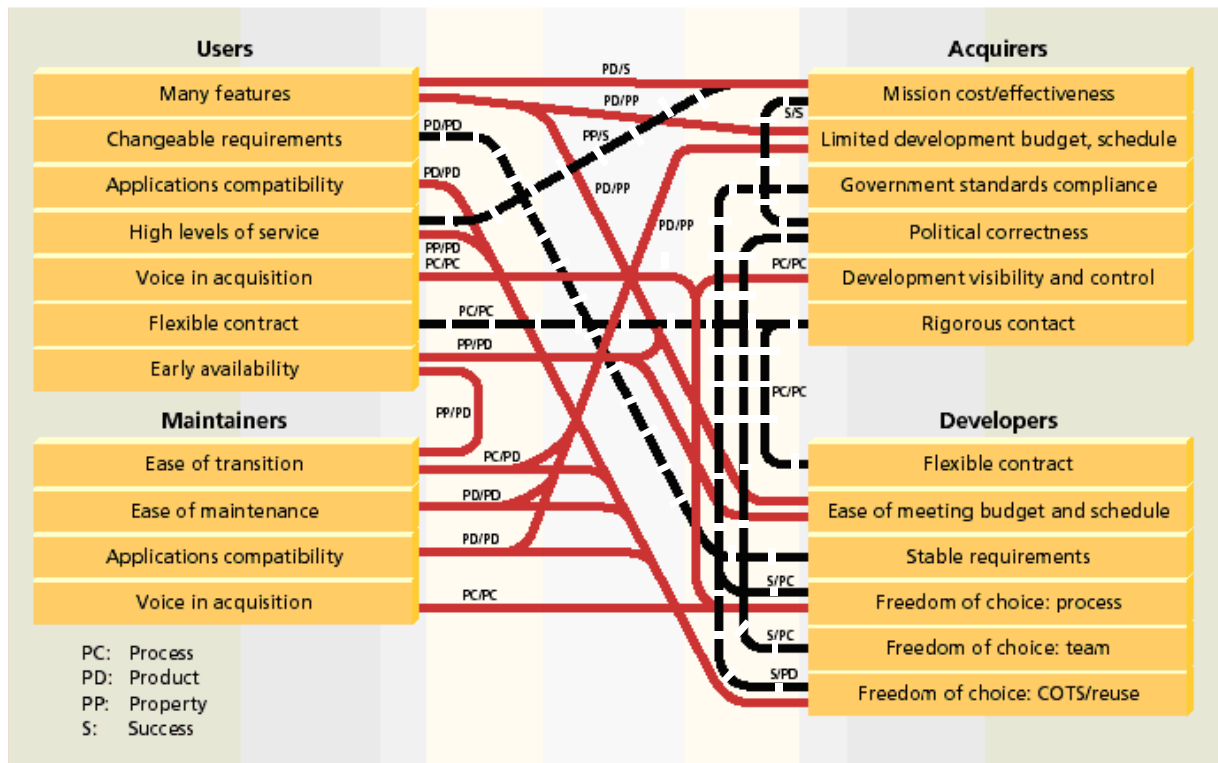
and resolution may facilitate more and better consideration of capabilities and quality attributes. In the short run, legacy maintenance and sustainment organizations that must deal with evolving requirements and re-engineering may benefit most from the use of semi-automated content analysis.

2.2 REQUIREMENTS ENGINEERING: MULTIPLE STAKEHOLDERS

Work elsewhere has suggested that requirements engineering in the Army can be hampered by the lack of information sharing and inter-organizational processes among combatant commanders, warfighter representatives in the Training and Doctrine Command (TRADOC), acquirers in program offices who transform user/customer requirements into contractual requirements, maintainers, and developers.³

Similar points have been made about commercial software environments. Figure 1 shows actual conflicts among stakeholder value propositions that were not resolved in a classic failed software project [Boehm 2000b, Boehm 2007]. The solid lines represent the Bank of America Master Net System project and the dashed lines show conflicts discovered in other failed projects [Boehm 2007]. The “S,” “PD,” “PC,” and “PP” annotations on the lines indicate whether a line reflected conflicts among the stakeholder’s *Success* criteria (e.g., verifiability, validity, and/or business case), *ProDuct* models (e.g., various ways of specifying operational concepts, ontologies, requirements, architectures, designs, and code, along with the interrelationships among them), *ProCess* models (e.g., waterfall or evolutionary), or *ProPerty* characteristics (e.g., cost, schedule, performance, reliability, security, portability, evolvability, or reusability tradeoffs). Although many of the interpretations would differ, a similar format and structure could be used in a military acquisition context.

³ Monarch, I. & Wessel, J. *Capabilities Engineering Framework: A Holistic Guide to Quality-Driven System of System Life-Cycle Engineering*, SEI technical report, forthcoming.



Modified from a graphic in "Future Challenges and Rewards for Software Engineers" [Boehm 2007]. Solid black lines were changed to dashed lines.

Figure 1: Stakeholder Perspective Differences in Commercial Software

Other work with the Army has focused mainly on the differences between combat developers (similar to user stakeholders in Figure 1) and material developers (similar to acquirer stakeholders in Figure 1).⁴ The commercial users want many features in their products that can conflict with the acquirers' cost and schedule success criteria. In the Army context, the problem is not so much that many features are wanted as gaining agreement and oversight on what features are needed and why they are needed.

The institutional Army is a very large enterprise made up of multiple organizations that must interact enough with each other and the operational Army in order to formulate, acquire, and field systems with the right operational capabilities. The Joint Capabilities Integration and Development System (JCIDS) is a Joint Chiefs of Staff policy that can be seen as a holistic attempt to address the problem of differences in perspectives between combat developers and material developers.

JCIDS is meant to promote clarification, prioritization, and traceability of operational and system requirements across the entire life cycle. Three separate documents are required: Initial Capabilities (ICDs), Capabilities Development Documents (CDDs) and Capability Production Documents

⁴ I. Monarch and J. Wessel. *Capabilities Engineering Framework: A Holistic Guide to Quality-Driven System of System Life-Cycle Engineering*, SEI technical report, forthcoming.

(CPDs).⁵ The ICDs, CDDs, and CPDs are meant to encourage incremental specification in order to avoid having requirements become outdated with respect to new threats and technologies.⁶

⁵ More detail about JCIDS, ICDs, CDDs and CPDs can be found in forthcoming SEI technical reports by Monarch and Monarch & Wessel.

⁶ See Appendix A for more detail about the JCIDS Key Performance Parameters.

3 Semi-Automated Content Analysis: Applying the Method

The results reported here are based on a semi-automated content analysis approach that combines detailed text analysis with semantic analysis done iteratively in collaboration with domain experts. Initial text analysis results were corroborated through interviews and other discussions. These results concern downstream activities such as testing, maintenance, and sustainment as well as activities further upstream that contribute to the development and capture of requirements. The aim of this research is to improve requirements specification by showing how problems and issues identified in downstream activities may be handled better and mitigated upstream.

Requirements specifications upstream and testing procedures and PCRs downstream are snapshots of ongoing work at the SEC. Particular attention is paid to the PCRs since they focus more directly on the problem areas faced by the SEC in its everyday work. In addition, many more of them are available for analysis.

The PCRs analyzed were text records derived from an existing SEC database. They were combined sequentially into three separate text files, one for each of the three participating projects. Each PCR is a single record in the SEC database. The records include several fields, which sometimes include values and sometimes are left unused. The values are all textual. Some are filled with a single term or phrase (e.g., dates, names of people, or short characterizations of status). Several other fields contain longer text with a paragraph or more of prose describing a problem, the considerations for resolving it, how it was resolved, and/or the pertinent rationales involved.

The text analysis described in this technical report was done using a tool called Leximancer.⁷ Leximancer automatically selects blocks of text, typically several sentences long, from the collection provided to it; however, its selection of the blocks can be constrained in various ways. The constraints specified for the current analysis guided the tool to select values based on the free-form text in the PCR and other form fields. The field names themselves were excluded, except when they were combined with symbolic values (e.g., status_closed). Boundaries between separate PCRs were respected, so that the tool's automatic selection of text blocks was prevented from combining text across the boundaries of adjacent PCRs.

The tool extracts concepts and themes from an analysis of co-occurrence of terms in the text blocks. The concepts are not simply literal terms but synonym lists consisting of terms used similarly in the blocks of text. Each concept is named by the most salient representative term in its respective synonym list.

Themes are collections of concepts whose meanings (represented as synonym lists) are closely associated with the other concepts that are collected into the same theme. Each theme is named by the concept most frequently connected to the other concepts in its respective cluster of concepts.

⁷ The product is described more fully at <http://www.leximancer.com>. As noted in Appendix B, many such tools exist, and they have different strengths and weaknesses. The SEI does not rank or promote them in any way.

A concept can be related with a concept in another theme, but it is less similar in usage to all of the other concepts in that theme than it is to the concepts within its own theme. Concepts also sometimes appear in the overlap of two or more contiguous themes.

The conceptual and thematic structure can be represented visually in concept maps, as shown in Figure 2 and Figure 3 with an example drawn from a semi-automated content analysis of CMMI® for Acquisition (CMMI ACQ) [Monarch 2008].

Much like Venn diagrams, the themes are represented spatially by colored circles (see Figure 3). Circle size is based on the spatial distribution of the concepts included in each theme. The brightness of each theme names represents the interconnectedness of the concepts included in it.

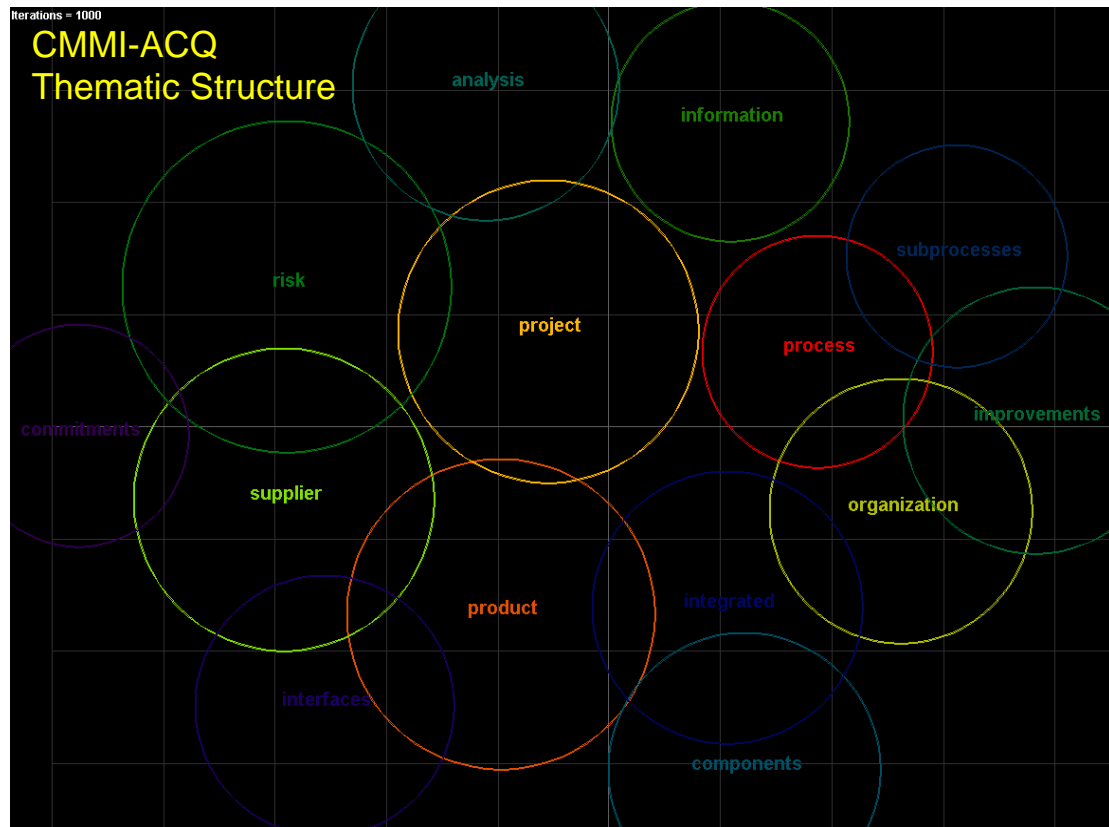


Figure 2: Example Concept Map Showing Themes

As shown in Figure 3, the concepts are indicated by dots. The size of the dots represents the interconnectedness of the terms in that concept's synonym set. The distance between the concept dots is a measure of how similar in usage they are to each other.⁸

⁸ Notice in Figure 3 that neither **customer** nor **users** is a central theme in CMMI ACQ. Rather, as pointed out by the arrows superimposed on the Leximancer figure, they are both concepts in the **product** theme. **Customer** also is a concept in the **supplier** theme. The authors of CMMI-ACQ did in fact choose to emphasize acquirer-supplier relationships. No one model can satisfy all perspectives.

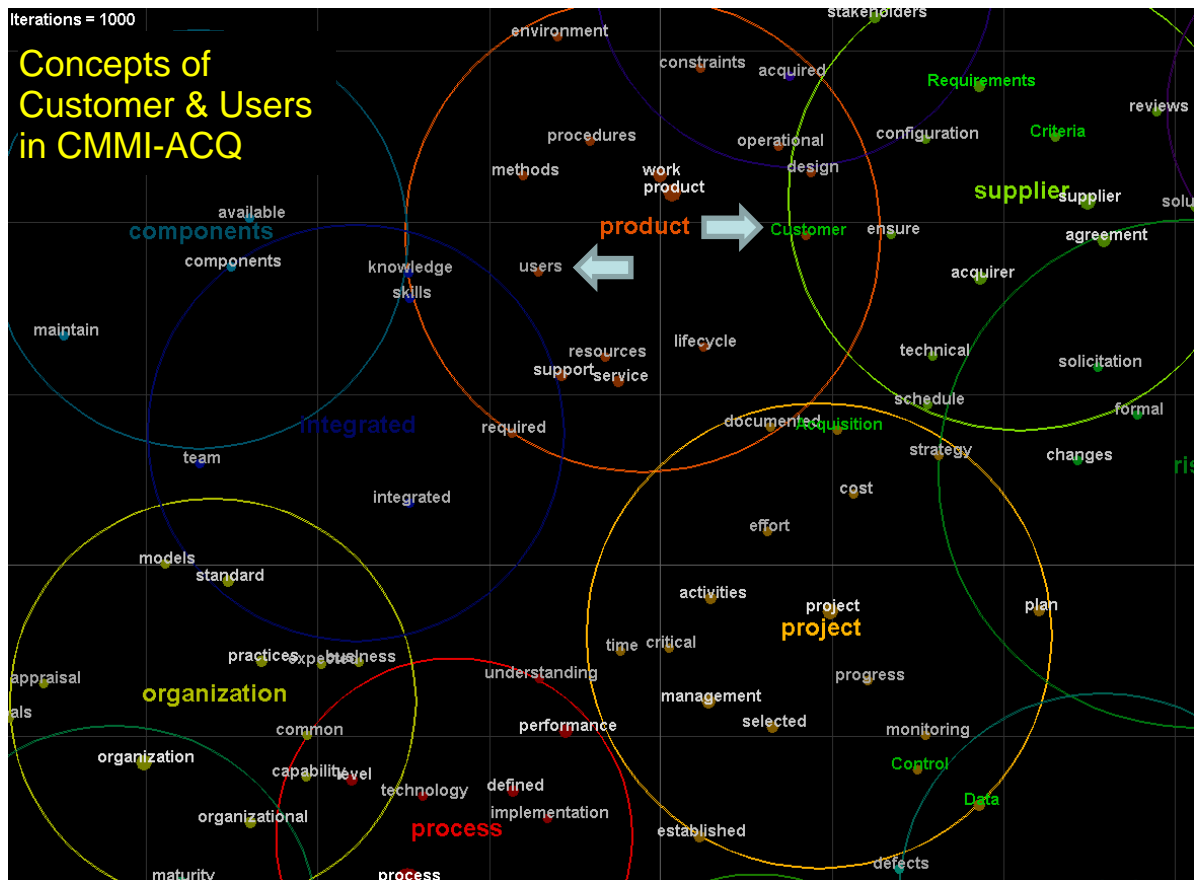


Figure 3: Example Concept Map Showing Both Concepts & Themes

As seen in Sections 4.4 through 4.6, the concept dots also can be linked by lines whose brightness represents the frequency of co-occurrence between each set of linked concepts. These kinds of co-occurrence can be especially useful as indicators of important causal relationships, especially across theme boundaries.

More detail about semi-automated content analysis can be found in Appendix B.

4 Analysis and Results

The results reported here are based on documents provided by three projects from one Army Software Engineering Center (SEC). Spanning the period from January 2005 through March 2007, the documents include requirements specifications, testing procedures, and PCRs from the three projects. Text and semantic analyses were done to characterize the overall meaning of all three sets of documents, with particular attention paid to the PCRs.

The Army practitioner-collaborators who participated in the study generally agreed with the interpretations identified by the initial text and semantic analyses. SEC project personnel already were aware of several issues and opportunities for improvement that were uncovered using the content analysis tools and techniques. Interviews and other discussions with project personnel identified a number of exemplary practices, some of which already were underway and others that were deemed worthy of being addressed by the SEC in the future.

Issues related to system usability constitute perhaps the most important semantic category supported by the automatically identified themes and concepts. That is true in spite of the fact that usability generally is not addressed explicitly as a quality attribute in the requirements specifications, test procedures, or PCRs. Nevertheless, topics related to information manipulation, user interfaces, and other factors important for operability and other aspects of usability are quite common. These usability-related topics were apparent in PCRs from all three projects, whether or not the projects were dealing with the introduction of new technologies. Not surprisingly, some of the PCRs that exhibited such user interface and usability issues did so in conjunction with other issues, such as reusability and maintainability, where important tradeoffs are often necessary.

4.1 REQUIREMENTS, VALIDATION, AND USABILITY AT THE SEC

While the requirements specifications and testing procedures do not explicitly contain context of use or acceptance criteria for usability, the SEC has in fact established processes to address usability-related issues. Operational scenarios and concepts have been developed by subject matter experts who are part of the SEC's maintenance and sustainment teams. These scenarios and concepts can serve as *de facto* specifications against which system changes can be validated.⁹ If explicitly documented, these operational scenarios and concepts may be suitable for future semi-automated content analyses.

Prima facie usability and context-of-use criteria would seem to be especially important for validation processes using measured attributes and for establishing traceability between system and user requirements. However, because of the SEC's current role (or lack of a role) in requirements development, they have been somewhat reticent to embrace quality attributes. The SEC is not always "involved in developing system requirements but rather in maintaining and sustaining the system according to these requirements." They agree that incorporating quality attributes into requirements would be "ideal when you are the prime developer, but when you are not, you really

⁹ This is similar to the role customer requirements play in CMMI *vis-a-vis* contractual product requirements.

cannot influence the trade space in deriving product requirements. You more or less have to accept what is given to you” [conversation with a member of the SEC].

Ideally, of course, maintainers/sustainers should be involved in the trade space because they—along with capability developers and material developers—are important stakeholders in capabilities and requirements development. However, even though circumstances may prevent the SEC from being involved in requirements development, quality attribute specifications captured in another document would provide operationalized or measured attributes that could facilitate maintenance and testing. Moreover, sustainment does involve refinements that also require gaining agreement of other stakeholders, even if it does not involve a formal requirements change. Although the trade space in these cases is not as large or heterogeneous, it still exists; refinements must still be validated.

The fact that the SEC has already begun to respond to issues and opportunities for improving interaction with users shows that validation with users is an important concern. For similar reasons, the SEC also has established Six Sigma groups to improve PCRs and determine whether requirements sufficiently inform testing procedures.

Overall, content analysis of PCRs and requirement specifications can be utilized to identify issues, opportunities for improvement, and potential exemplary practices. This will be seen in greater detail later in this section, especially with respect to the role of usability in the PCRs and the need for operationalizing it in requirements specifications or in other kinds of documentation.

Though the SEC has processes, software design criteria, and coding conventions to correct offending code when it is found during routine maintenance procedures, it may be that there are opportunities for improving these processes. New ways of interacting with other stakeholders to define quality attributes may be important in achieving such improvement. In addition to usability, quality attributes such as modifiability, reusability, interoperability, and other quality attributes emphasized in ISO standards and the software architecture literature may cover other issue areas that further analysis of additional PCRs and other documentation might uncover. Such analyses may prove to be viable additions to existing causal analysis and resolution processes.

4.2 AN INITIAL SEMANTIC CATEGORIZATION

An initial set of rough semantic categories (see Figure 4) were crafted based on numerous iterations of the automated text analyses.¹⁰ The terms naming the categories are not necessarily used in the PCRs themselves. However, they represent or classify themes that were automatically derived from all three projects’ PCRs in categories whose meanings all SEC members share. These categories are used to enable comparison and contrast of the analysis results across the three different projects. The meanings of these categories remain informal; however, no problems with their meaning or applicability across the projects were raised by the SEC personnel who collaborated in the analysis.

¹⁰ The iterations are necessary to settle on the best level of abstraction for the concept maps and for experimentation with manually added seed concepts. See Sections 4.5 and 4.7, and Appendix B for more detail about the semantic analysis.

- Information Manipulation, User Interface, and other Usability Factors
- Hardware System or Modules containing or controlled by information technology or software
- Context of Use (Mission, Exercise, Training, User)
- Testing and Maintenance, Configuration Management
- Software, Software System, Data, Data Standards

Figure 4: Semantic Categories

The semantic categories were created initially by examining the concept maps from all three projects and noting patterns of similar topics suggested by the terminology used. This was corroborated by reading the text indexed by the concepts that name the themes. More than one of these concepts clearly fit under a single higher level semantic category, and some of these same concepts fit under more than one of the higher level semantic categories. That is why the same concepts are associated with more than one semantic category in Figure 12 and Figure 19.

The current categorization should undergo more elaboration and formalization over time. This may be particularly valuable for better situating, understanding, and using quality attributes in the local SEC context. It also may provide a basis for computational support for assigning and characterizing such quality attributes.

4.3 QUALITY ATTRIBUTES IN THE THREE PROJECTS

Project 1 requirements specifications and testing procedure documentation did not contain usability scenarios, objectives, or ranges of quantified system responses deemed acceptable in interaction with the user. However, operational scenarios and, in some cases, quantified inputs and outputs are provided for other quality attributes including accuracy, security, and data handling. There are no such criteria for usability in the Project 1 requirements specifications and testing procedures that have been analyzed thus far. There are no usability criteria consisting of measurable attributes for suitability or usefulness in the context of use, learnability, or what counts as disruption for warfighters interacting with the system. This is the case even though decisions where such criteria had to be determined on the spot and acted upon were reported in the Project 1 PCRs during the same period.

Project 2 requirements and test documentation considered speed of employment, accuracy, security, data handling, situational awareness, and interoperability. Similar to Project 1, operational scenarios or quantified inputs and outputs sometimes are provided for accuracy, security, and data handling. However, usability criteria containing measureable attributes are not included in the requirements or test documents, even though they again come up in the PCRs. Usability criteria for situational awareness that would operationalize what is meant, for example, by rapid acceptance or prioritization of large amounts of data from a variety of digital networks are not covered in the requirements and test documents.

Similarly, Project 3 requirement specifications consider survivability, system responsiveness, data handling, security, reliability, availability, and maintainability. There are some scenarios and quantified inputs and outputs for survivability, system responsiveness, and data handling; however, scenarios, quantified inputs, and outputs rarely exist for security, reliability, maintainability,

and availability. Though once again, usability criteria are not explicitly stated in Project 3 requirements specifications.¹¹

4.4 PROJECT 1

The results from Project 1 are based on 647 PCRs. They were automatically apportioned for analysis into a total of 1103 text blocks. As seen in Figure 5, each word or words in the right-hand column names both a theme and the concept after which it is named by the automated text analysis. The number in parentheses to the right of each one represents the number of text blocks in which the concept corresponding to the theme with the same name occurs, including all of the terms in its synonym list. Additional terms occasionally follow in parentheses for further clarification. The six semantic categories described in Section 4.2 are listed in the left-hand column of Figure 5.

Semantic Categories	Number of Text Blocks where Theme-Naming-Concepts Occur
Information Manipulation, User Interface, and other Usability Factors	Azimuth (36) (enter), reset (98), screen (501), send (161), displayed (390) (order of buttons), data (190), security (16)
Hardware System or Modules containing or controlled by information technology or software	Computer (71), r-pda (1103) (Ruggedized-Personal Digital Assistant), gun (244), shut_down (25) (HW)
Context of Use (Mission, Exercise, Training, User)	Security (11)
Testing and Maintenance, Configuration Management	mailto (17), Srs_19 (115), limits (16) (set by requirements)
Software, Software System, Data, Data Standards	Computer (71), r-pda (1103) (Ruggedized-Personal Digital Assistant), data (190), reset (98) (required), FOS (303) (Forward Observer System), azimuth (36), security (11), send (161), shut_down (25) (software)
Systems and Software Engineering	SRS_19 (115) (Software Requirements Spec)

Figure 5: Semantic Categories in Project 1

¹¹ No testing procedure documentation was provided for analysis by Project 3.

All six semantic categories are represented; however, the user interface and usability category provides the most insight into the types of problems that are exhibited in this set of PCRs. The user interface and usability factors also overlap notably with the hardware and software categories. In this case, the themes identified by the text analysis tend to belong to more than one semantic category. The PCRs cover a time period when the project was still adjusting to issues arising from reusing software developed for a desktop computer in a ruggedized pda with a much smaller screen, compacted controls, and touch screen interaction.¹²

Figure 6 and Figure 7 contain the concept maps from the text analysis of the Project 1 PCRs. Figure 7 displays all of the concepts identified up to but not necessarily reaching the limit of 300 concepts set for this analysis.¹³ The concepts are fairly evenly distributed among themes; however, there are quite a few overlapping themes with the same concepts in more than one theme.

¹² This circumstance led to the **r-pda** concept in this collection of PCRs having conceptual traces in *all* of the text blocks.

¹³ Because of the number of densely pack concepts, some of the concept names are difficult to see in Figure 7.

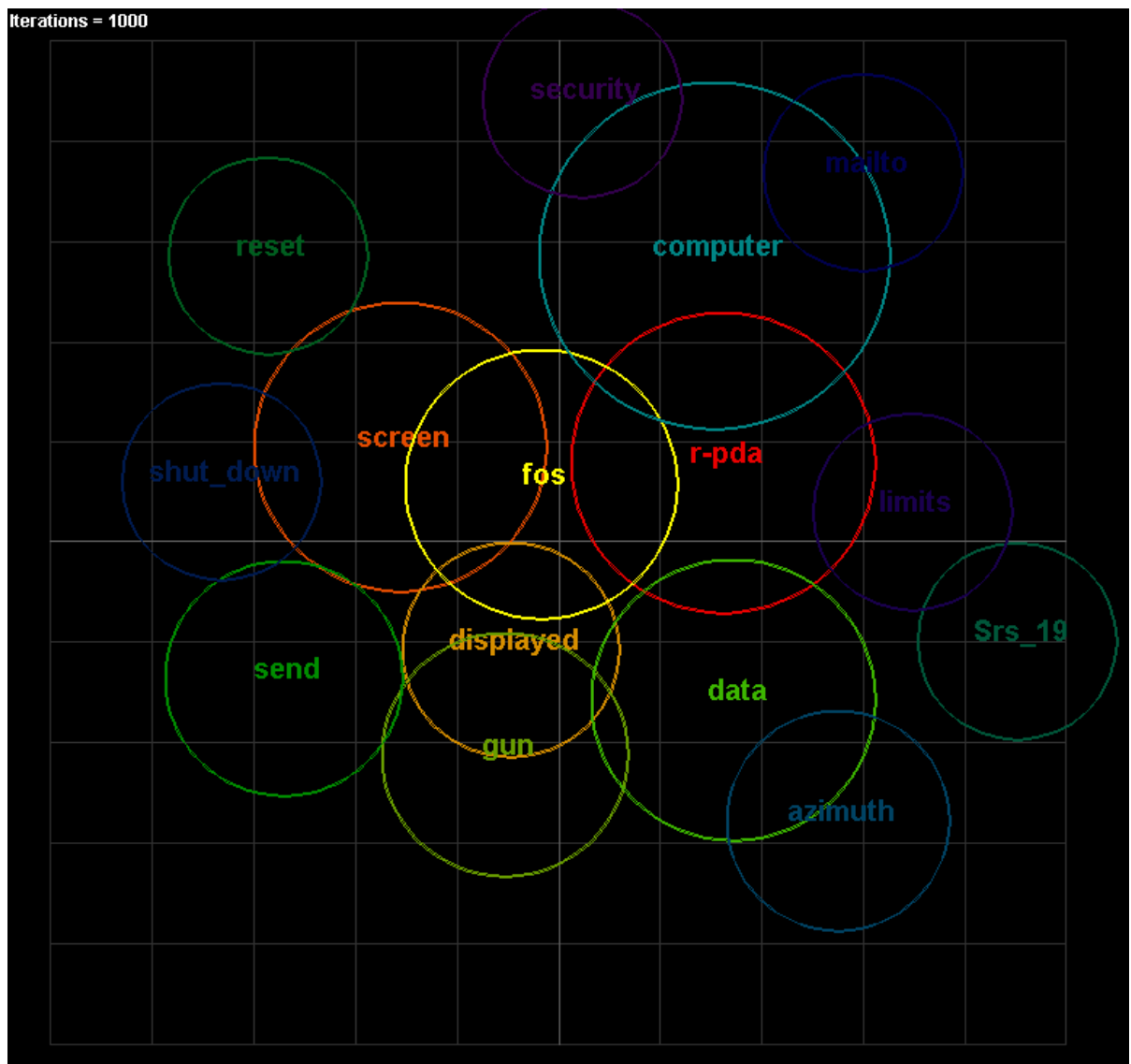


Figure 6: Project 1 PCRs Concept Map Showing Themes Only

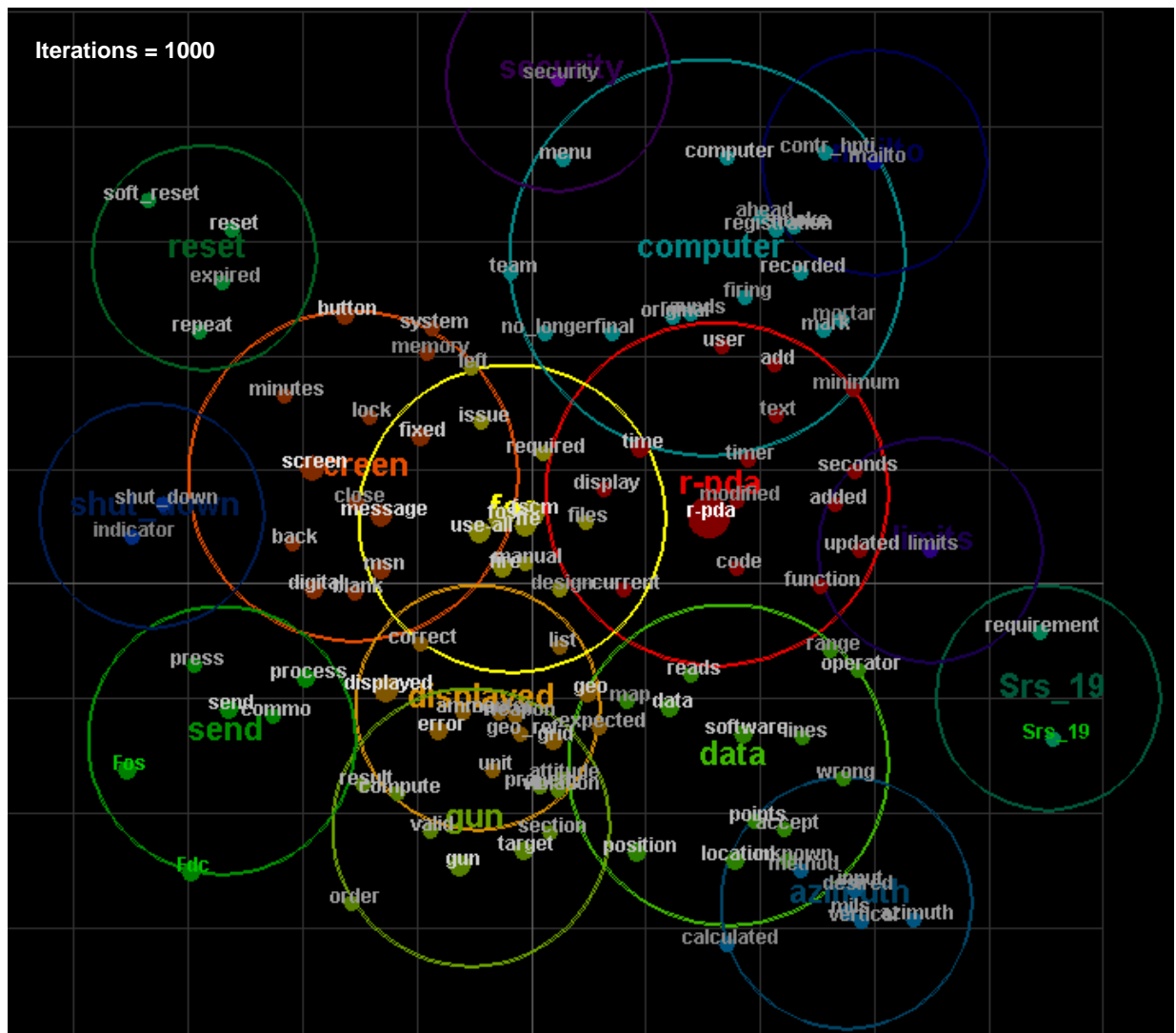


Figure 7: Project 1 PCRs Concept Map Showing Both Concepts & Themes

As mentioned, the software and hardware categories overlap with the user interface and usability factor themes. For example, Figure 8 shows linkages among very frequent concepts stemming from the user interface concept **screen** linking to the named concepts **FOS** (Forward Observer System), **FFE** (Fire for Effect), **FSCM** (Fire Support Coordination Measures), **message**, **r-pda**, **button**, **gun**, **send**, **displayed**, **error**, **geo**, **fire**, and **data**.¹⁴ These concepts are at the heart of what is being discussed in the Project 1 PCRs; they identify important issues and possible opportunities for improvement at the intersections between hardware, software, user interface, and usability.

¹⁴ Concept and theme names are shown in bold face type for emphasis and to distinguish them from ordinary usage of the same words throughout the remainder of the text and figure captions in this section.

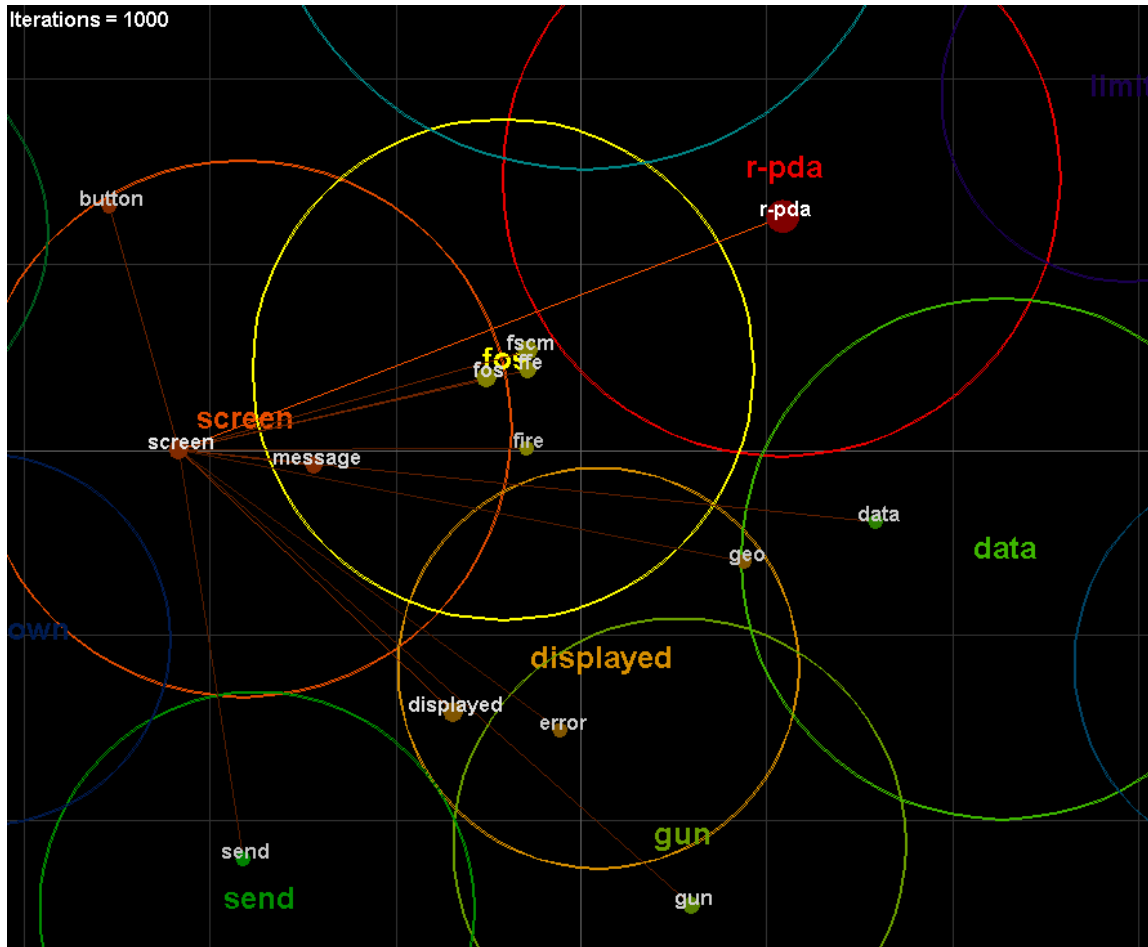
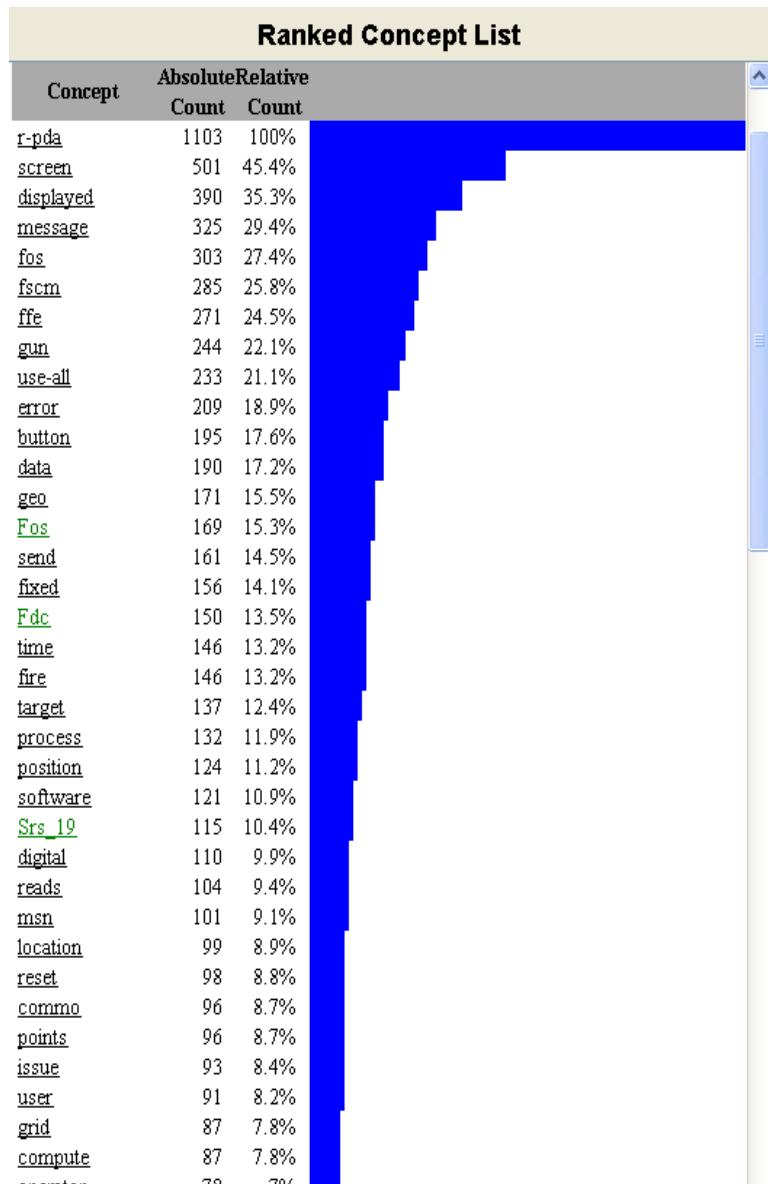


Figure 8: Links Among Key Concepts Derived from Project 1 PCRs

Figure 9 shows a ranked concept list on the left and a ranked concept co-occurrence list on the right. The ranked concept list includes the most common concepts, expressed in actual numbers and percentages of the total text blocks where they occur. The concept co-occurrence list provides further detail about the concepts related to **screen**, in order of their frequency of co-occurrence. The absolute count is the number of text blocks in which **screen** and each of the other concepts co-occur. The relative count is the percentage of text blocks where **screen** co-occurs with each other concept as a proportion of the total number of text blocks where screen co-occurs. The number of different concepts linked to **screen** (112) is given in parentheses at the top of the list.

The top-ranked concept and the most interlinked concept with **screen** is **r-pda**, which involves both hardware and software systems. Concepts in both these semantic categories are quite frequent and connected, but so are **screen** and other user interface- and usability-related concepts such as **displayed**, **message**, **button**, **data**, **geo**, and **send**. In fact, as is the case with **screen**, every one of these concepts is related to the other frequently used concepts identified by the text analysis, albeit with different strengths of connection.

The concept **error** in the **displayed** theme presents an interesting case (see Figure 10). One of the issues indexed by **error** is whether users are provided proper feedback for data entry errors. Notice that **error** links to **button** on the upper left part of the circumference of the theme **screen** in Figure 10.



Concepts Related to Screen (Count: 112)

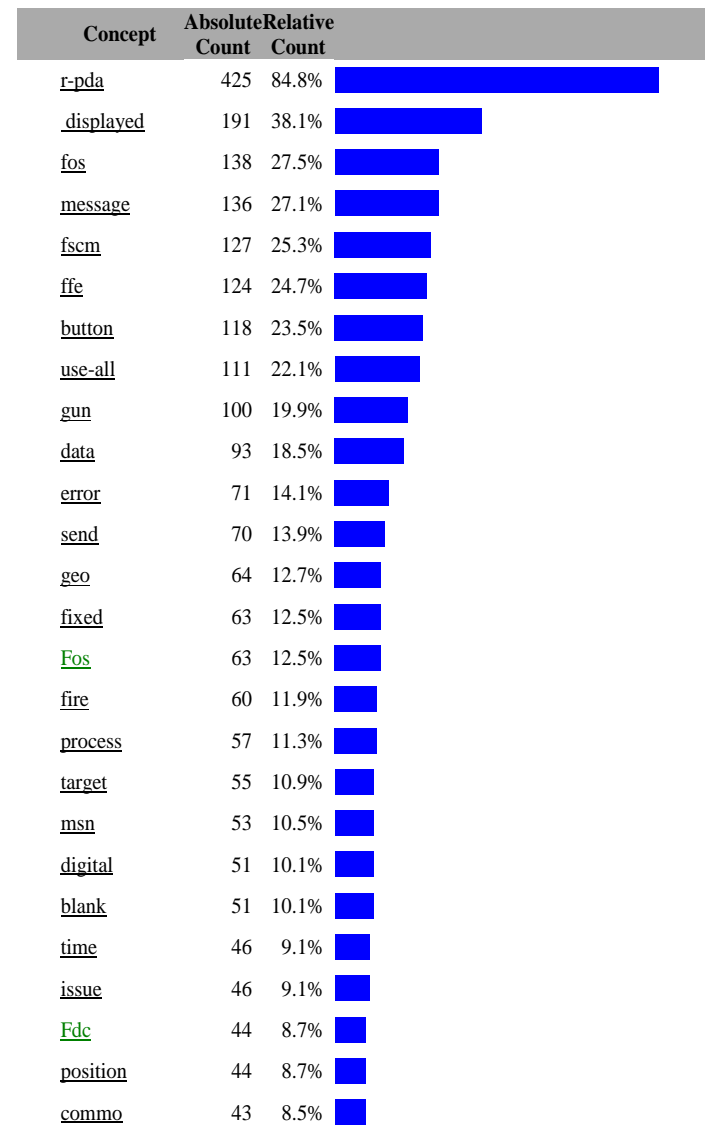


Figure 9: Project 1 PCR's Ranked Concepts and Concepts Related to Screen

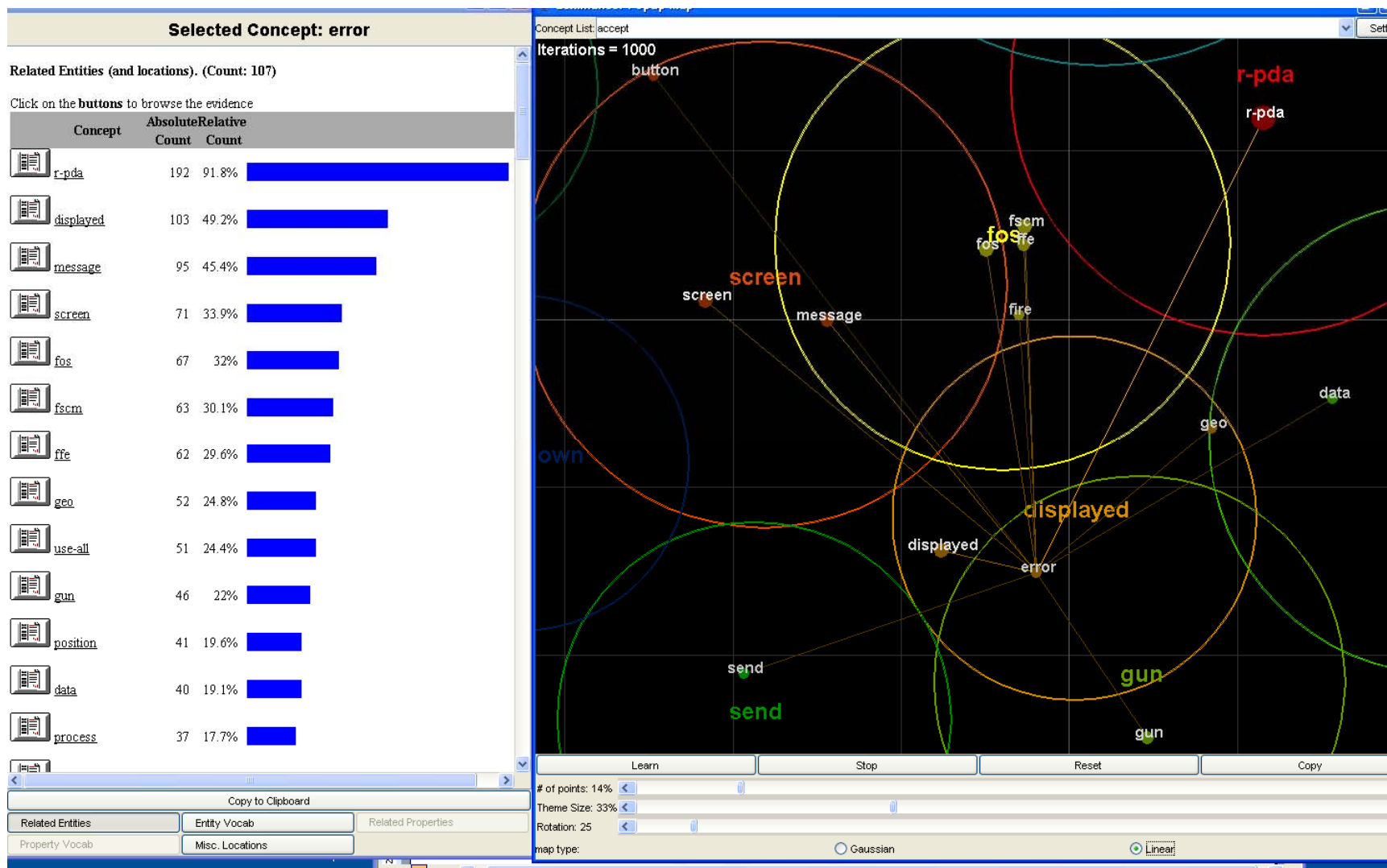


Figure 10: Many PCRs Point to Issues Providing Feedback with Respect to Data Entry Errors

As seen in Figure 11, **button** also links to many other concepts, most prominently to **r-pda**. However, **button** also links frequently to other user interface concepts, in descending order of frequency: **screen**, **displayed**, and **message**, followed by the system and software modules **FOS** and **FSCM**, a context of use concept (**FFE**), another user interface item (**use-all**), and finally to hardware employed in the context of use, namely **gun**. These linkages among concepts, even to a layperson, are clearly about the pda's use in controlling the firing of a gun. A further link, to the left lower part of the circumference of the theme **gun**, would seem to indicate issues in the pda's role in carrying out gun orders. These are the contexts in which feedback regarding data entry errors has been an issue.

Notice also in Figure 11 that the text associated with any of the concepts can be seen by clicking on one of the "buttons to browse the evidence." By reading the five text blocks (not shown in the figure) in which the concepts **order** and **button** appear together reveals another issue. This issue concerns the ordering of buttons on the computer menu, their accessibility, and their usability.

In under a half hour of perusing the concept map and using the browsing facilities, someone who knows how to use the text analysis tool but is only slightly familiar with the domain is able to identify two kinds of usability issues: (1) identification and characterization of data entry errors and providing appropriate feedback, and (2) proper layout and operability of soft buttons on a small PDA screen in a context where reusing software adapted from a system using a desktop computer with a larger screen and keyboard. While this is not an exhaustive use of the intelligent browsing and smart search procedures enabled by the content analysis tool, it is clear that a more complete classification of usability issues is well within reach. The same may be true for other quality-attribute-related issues such as reusability, maintainability, and reliability.

By devoting a relatively short amount of time to learning about the same kinds of semi-automatic content analysis methods and tools, practitioners and other subject-matter experts should be able to begin doing similar analyses on their own. With sufficient guidance, they also should be able to integrate such methods with their existing causal analysis and related processes.

4.5 PROJECT 2

The results from Project 2 are based on PCRs that cover roughly the same time period as those from Project 1. They were apportioned for analysis into a total of 958 approximately equally sized text blocks. Once again, each word or words in Figure 12 names a theme and the concept after which it is named by the automated text analysis, and the number in parentheses to the right of each one represents the number of text blocks in which the concept occurs. Additional terms follow in parentheses for further clarification in one instance. The other five semantic categories are represented in the figure; however, none of the themes identified for this project by the text analysis tool focus on the “systems and software engineering best practices” semantic category. Not surprisingly in a maintenance organization, the concept **Discovering_Activity_Testing** occurs pervasively in all 958 text blocks.

Semantic Categories	Number of Text Blocks where Theme-Naming-Concepts Occur
Information Manipulation, User Interface, and other Usability Factors	Message (246), User (152), appears (27), Q_C_A_Usability (635), charge (15) (as displayed)
Hardware System or Modules containing or controlled by information technology or software	Network (21), cable (78), guns (353), drive (31), round (12), charge (15)
Context of Use (Mission, Exercise, Training, User)	user (152)
Testing and Maintenance, Configuration Management	Discovering_Activity_Testing (958), failure (58), problem (392), Q_C_A_Usability (635), dry_run (499)
Software, Software System, Data, Data Standards	software (107), SW (106)
Systems and Software Engineering	None

Figure 12: Semantic Categories in Project 2

In addition to providing software for Project 1, Project 2 continued its sustainment of similar software for a desktop computer. Recall that Project 1 had to reuse and adapt software for a significantly different computer environment. Project 2 has not faced the same challenges; however, usability-related issues arise in Project 2 as well.

As with Project 1, two figures contain the basic concept maps from the text analysis. Figure 13 shows the themes only, and Figure 14 also displays all the concepts identified up to but not necessarily reaching the 600 concept limit specified for the automated text analysis of the Project 2 PCRs. Notice in Figure 14 that a very large number of related concepts are associated with each other in a much smaller number of themes, which themselves overlap noticeably. The semi-

automated content analysis helps identify and clarify a thematic structure that would be difficult if not impossible to recognize by reading only the detailed text.

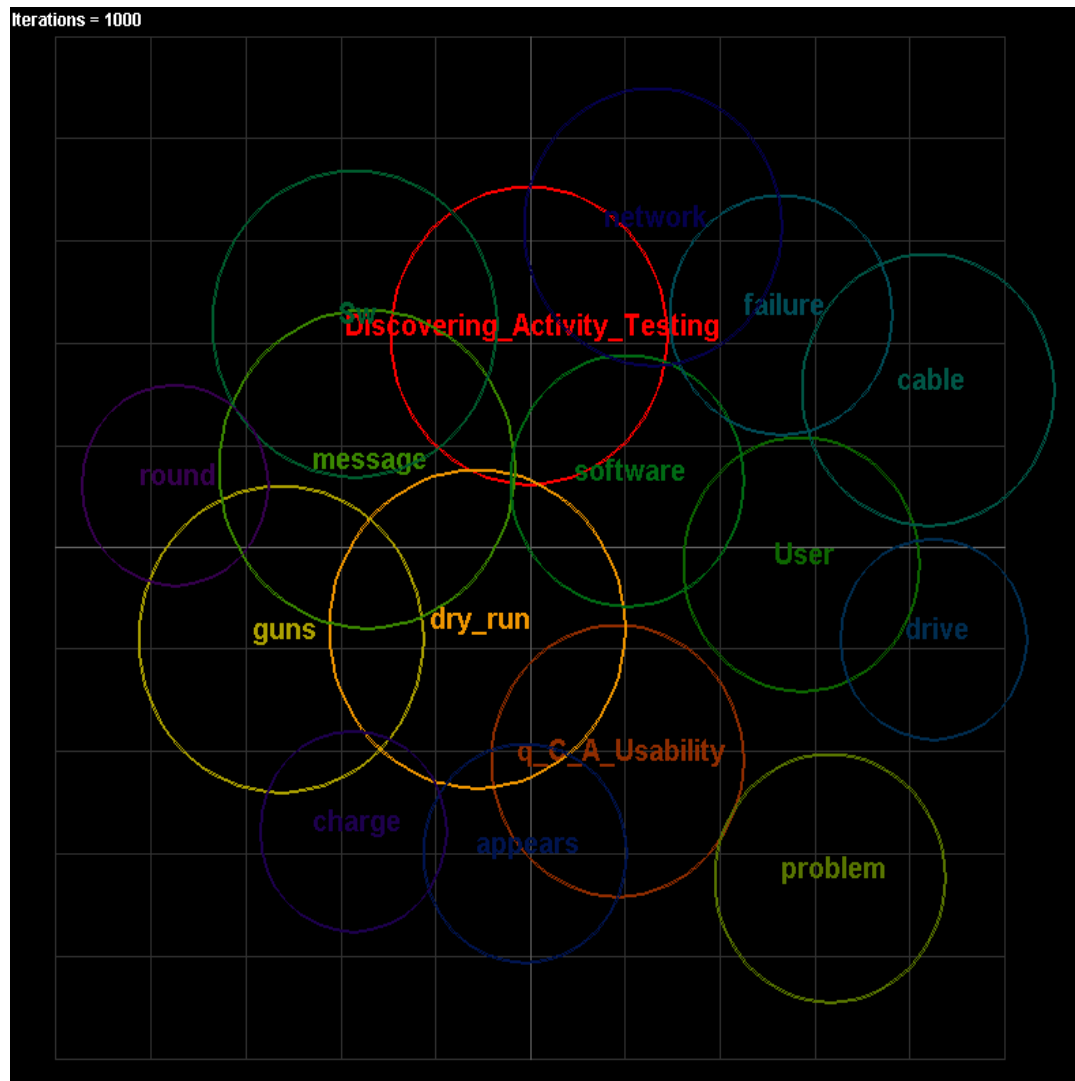


Figure 13: Project 2 PCR's Concept Map Showing Themes Only

Note that the concept limit increased in the analysis of PCRs for this project. Several new seed concepts also were added manually to those generated automatically.¹⁵ The manually generated seed concepts included the creation of alias codes so proper names could not be traced back publicly to particular individuals. More importantly, seed concepts allow an analyst to semantically

¹⁵ Experimentation can be done iteratively. The Leximancer tool identifies seed concepts in the initial stages of processing and then weeds some of them out and adds new ones as it determines which terms have the most salient co-occurrence profiles. In subsequent runs, analysts can add seed concepts they want examined as candidate concepts. These need to be built from and associated with terms (in synonym lists) that exist in the texts being analyzed. Sometimes these are weeded out as well, but the automated analysis can designate them as concepts and sometimes themes.

combine fields and values that are not recognized automatically, such as **Q_C_A_Functionality** and **Q_C_A_Usability** (Q_C_A was abbreviated from Quality_Characteristic_Affected).

A similarly named PCR field, “Quality_Characteristic_Affected,” had been used by this project, but it was discontinued during the time period covered by this analysis. The two seed concepts just mentioned were introduced by the authors to probe further about the use of that field. Although some other field/value combinations, such as **Discovering_Activity_Testing** and **Status_List_Closed**, were automatically recognized, Q_C_A as a field was not recognized automatically with any of its values. Aside from “usability” and “readability,” “functionality” and “reliability” were the only other values entered in the Q_C_A field.¹⁶ A value for Q_C_A was specified for only 57 of the 567 PCRs analyzed for Project 2. However, the analysis found that Q_C_A was in fact associated with very highly connected concepts in the project’s PCRs.

The retirement of the “Quality_Characteristics_Affected” field may have been premature. The question as to what quality characteristic is affected is, after all, still used in the project’s peer reviews. Although the field was only used for a short time, its retirement may have been due to inadequate support of quality attribute selection and articulation.

One other seed concept, “popup,” was created manually. That was due to the fact that **pop** was recognized automatically, but “up” was not, even though “pop” never occurred without being followed by “up” in the PCRs. Interestingly, while **popup**, **Q_C_A_Functionality**, and **Q_C_A_Usability** all were included in the final concept map, only **Q_C_A_Usability** was designated as a theme.

As with Project 1, themes that appear to be related to user interface and usability-related issues are prevalent. As seen in Figure 14 there also was considerable conceptual density.¹⁷ Figure 15 shows linkages between the manually created concept **Q_C_A_Usability** and other highly ranked key concepts that were derived from the automated text analysis of the Project 2 PCRs.

¹⁶ Q_C_A_Readability was added to **Q_C_A_Usability**’s synonym list. Since reliability only occurred once, it was not useful as a probe.

¹⁷ The concepts are not meant to be readable in this figure.

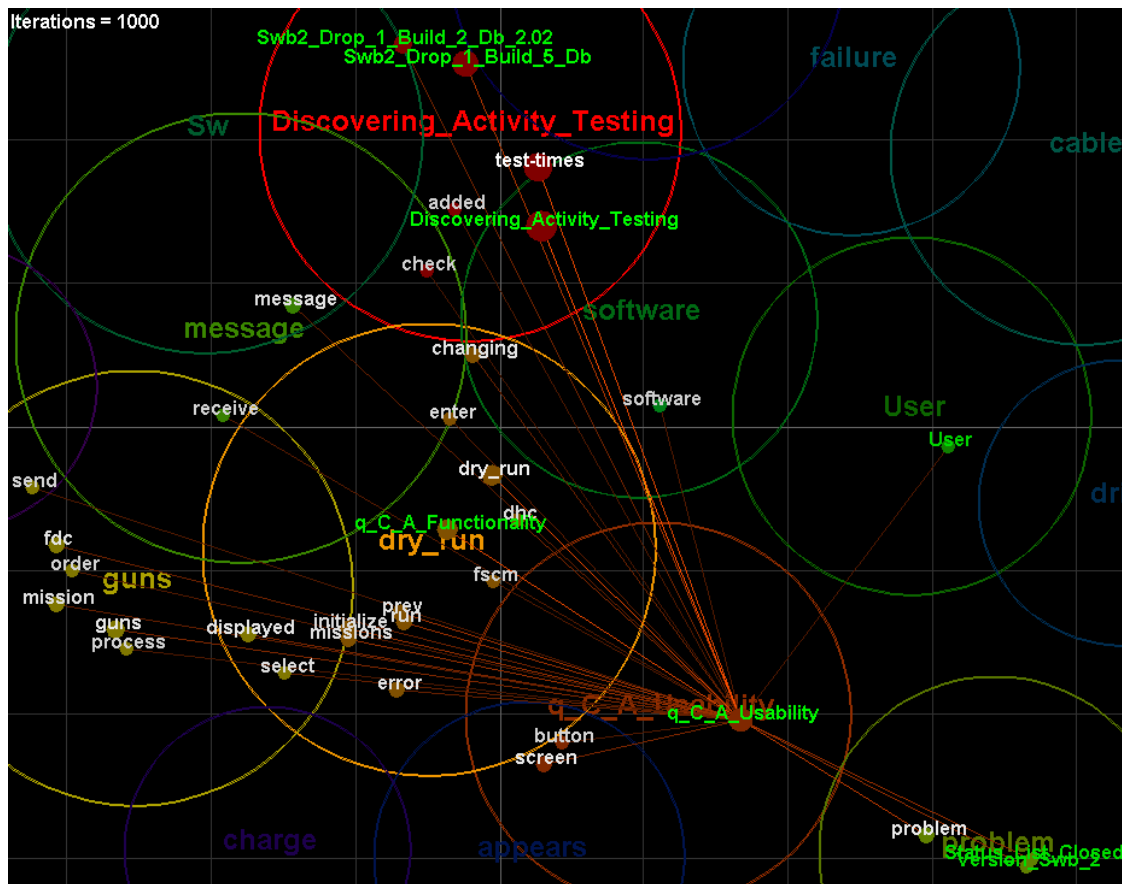


Figure 15: **Q_C_A_Usability** Concept Links

The list on the left side of Figure 16 shows quantitative results for the most frequently occurring concepts found by the automated text analysis. **Q_C_A_Usability** itself is fourth on the ranked list of the most frequent and connected concepts. Moreover, one third of the top 33 concepts listed there are user interface or usability related, and they are strongly linked to **Q_C_A_Usability**. The other two thirds of the top 33 concepts on this list also are reasonably strongly linked to **Q_C_A_Usability**.

Ranked Concept List

Concept	Absolute Count	Relative Count
<u>Discovering Activity Testing</u>	958	100%
<u>test-times</u>	866	90.3%
<u>Swb2 Drop 1 Build 5 Db</u>	749	78.1%
<u>q C A Usability</u>	635	66.2%
<u>dry run</u>	499	52%
<u>q C A Functionality</u>	422	44%
<u>Swb2 Drop 1 Build 2 Db 2.02</u>	421	43.9%
<u>problem</u>	392	40.9%
<u>screen</u>	370	38.6%
<u>guns</u>	353	36.8%
<u>displayed</u>	325	33.9%
<u>error</u>	287	29.9%
<u>fdc</u>	286	29.8%
<u>changing</u>	270	28.1%
<u>prev</u>	254	26.5%
<u>message</u>	246	25.6%
<u>mission</u>	236	24.6%
<u>missions</u>	233	24.3%
<u>initialize</u>	218	22.7%
<u>Status List Closed</u>	218	22.7%
<u>Version Swb 2</u>	217	22.6%
<u>select</u>	208	21.7%
<u>enter</u>	198	20.6%
<u>dhc</u>	178	18.5%
<u>User</u>	152	15.8%
<u>receive</u>	141	14.7%
<u>button</u>	137	14.3%
<u>send</u>	134	13.9%
<u>order</u>	132	13.7%
<u>added</u>	128	13.3%
<u>fscm</u>	125	13%
<u>process</u>	125	13%
<u>press</u>	124	12.9%
<u>check</u>	120	12.5%
<u>software</u>	107	11.1%
<u>Sw</u>	106	11%
<u>time</u>	106	11%
<u>data</u>	104	10.8%
<u>fix</u>	95	9.9%
<u>position</u>	94	9.8%
<u>work</u>	91	9.4%
<u>code</u>	90	9.3%

Related Entities for Selected Concept Q_A_C Usability. (Count: 247)

Concept	Absolute Count	Relative Count
Discovering Activity Testing	457	71.9%
dry run	417	65.6%
test-times	390	61.4%
q C A Functionality	385	60.6%
problem	306	48.1%
Swb2 Drop 1 Build 5 Db	279	43.9%
prev	234	36.8%
missions	224	35.2%
Status List Closed	218	34.3%
Version Swb 2	217	34.1%
initialize	215	33.8%
dhc	173	27.2%
guns	163	25.6%
screen	161	25.3%
displayed	154	24.2%
Swb2 Drop 1 Build 2 Db 2.02	141	22.2%
changing	135	21.2%
fdc	124	19.5%
error	113	17.7%
select	106	16.6%
message	106	16.6%
mission	103	16.2%
enter	95	14.9%
receive	78	12.2%
User	67	10.5%
fscm	66	10.3%
software	64	10%
cpp	64	10%
order	63	9.9%
button	62	9.7%
fire	61	9.6%
added	60	9.4%
send	60	9.4%
build	57	8.9%
data	57	8.9%
process	53	8.3%
check	52	8.1%
fix	48	7.5%
work	47	7.4%
press	47	7.4%
system	45	7%
code	44	6.9%

Figure 16: Ranked Concept List & Text Block Counts for Concepts Co-Occurring with Q_C_A Usability

The list on the right side of Figure 16 shows the rank order of the concepts that co-occur most frequently with **Q_C_A_Usability**.¹⁸ In fact, the co-occurrence of the top twenty-eight of these concepts with **Q_C_A_Usability** ranges from 10 percent to over 70 percent. Although maintenance and testing clearly are the main focus of the Project 2 PCRs, user interface and usability factors are also very significant, especially as seen through the conceptual linkages with **Q_C_A_Usability**.

Several concepts related to user interface and usability factors are collected in the **Q_C_A_Usability** theme, particularly when the **Q_C_A_Usability** theme overlaps with the **appears** and **dry run** themes. One term in the **Q_C_A_Usability** theme, **stack_dump**, stands out as being quite different from all the other concepts in that theme. Figure 17 shows a concept map with links radiating from **stack_dump**. Figure 18 contains further detail about the situation in which a stack dump can occur.¹⁹ A stack dump appeared when a user moved a mouse over a certain point on the screen under the rare conditions when a “divide by zero” error could occur. The stack dump consisted of a pop up dialog indicating the lines of code, file, and function that were executing when the error occurred. In test situations, the system would either reboot or lock up after the error and stack dump display. Such information obviously is useful for maintainability of the software. However, as noted in one of the PCRs, such behavior should not occur when errors like this are encountered in the field. A maintainability feature would then cause a disruption and interfere with usability or availability.

¹⁸ Recall that the ranking is based on how many text blocks are shared with **Q_C_A_Usability**.

¹⁹ Proper names are blanked out such that specific individuals are not identified by name.

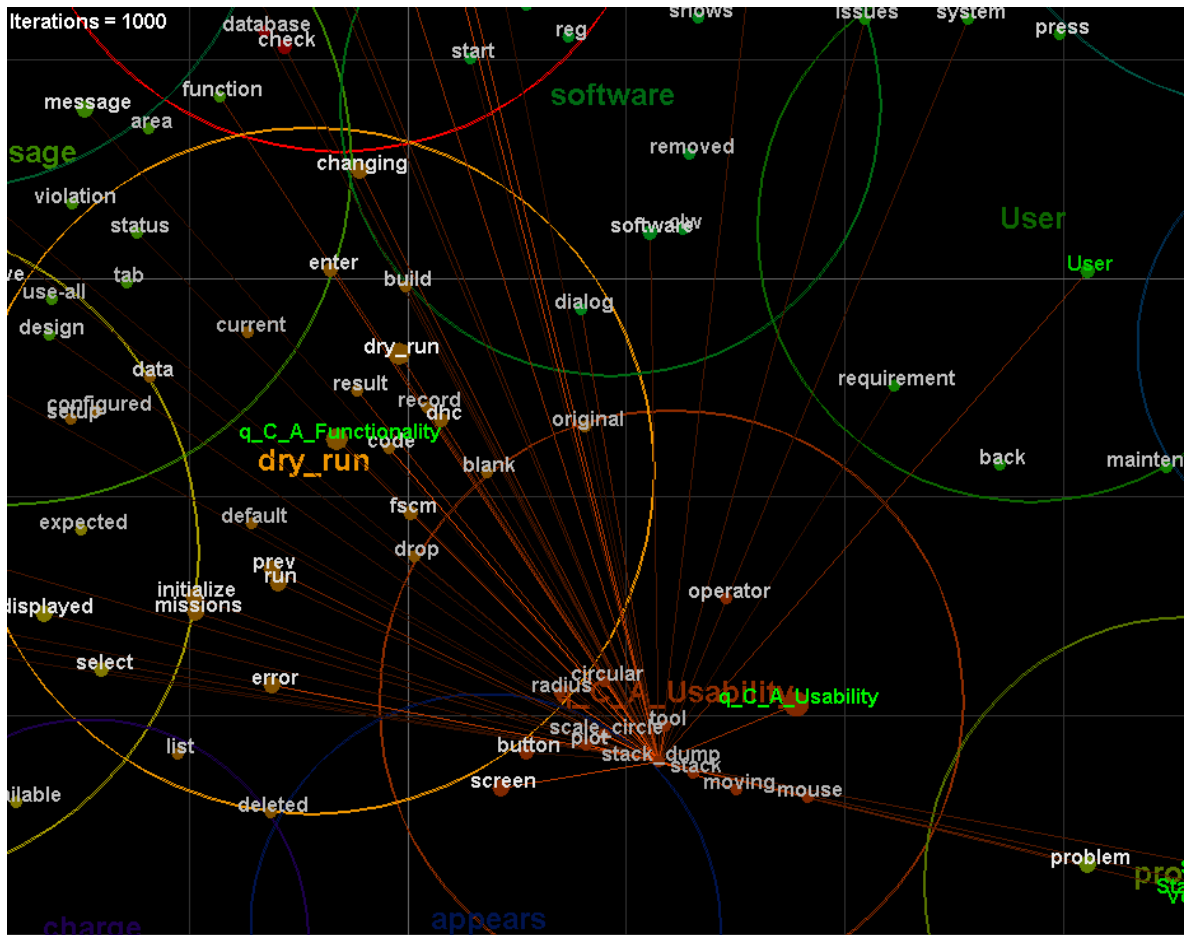


Figure 17: Less Frequent Concepts Collected in the Q_C_A Usability Theme – Focus on **stack_dump**

Issues of this kind are quite rare at the SEC, and even more so in their field trouble reports which are exceedingly rare. The stack dump PCRs were written while the product still was being refined.

The example is entirely atypical. It is shown here only since it is a good illustration of how different quality-attribute-related indicators can co-occur. As seen in Figure 18, it also provides a good example of how an analyst can use smart searches to traverse through a voluminous amount of information to scan only the pertinent text during causal analysis.

While “usability” was the value entered in the PCR’s “Quality_Characteristic_Affected” field, the situation is more complicated. Two quality attributes are involved: usability and maintainability. The necessary corrective actions were taken by the SEC, but tradeoffs of this kind might benefit from explicit consideration of quality attributes and their possible impacts on each other. This may be especially important with respect to usability. Features that support maintenance and sustainability can be evaluated upfront to anticipate their effects elsewhere. Errors happen; processes for how they should be handled under different conditions can be established upfront in collaboration with the stakeholders most likely to be affected.

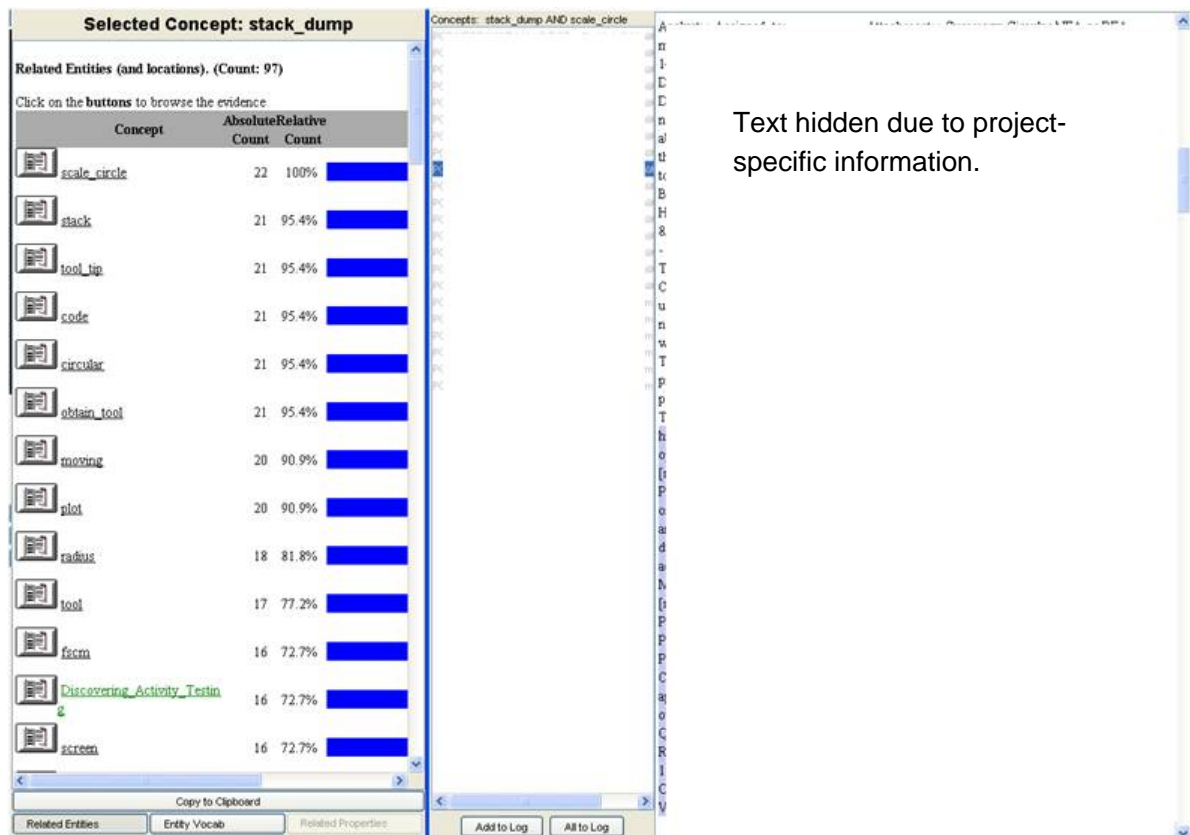


Figure 18: Browsing a PCR with the Co-occurring Concepts **stack-dump** & **Q_C_A_Usability**

4.6 PROJECT 3

From project 3, 550 PCRs were apportioned for analysis into a total of 2445 approximately equally sized text blocks. Each word or words in the right-hand column of Figure 19 names both a theme and the concept after which it is named by the automated text analysis. The number in parentheses to the right of each represents the number of text blocks in which the concept occurs, including all of the terms in its synonym list. Additional terms again follow in parentheses for further clarification. Five of the six semantic categories are represented in the figure for Project 3; however, in this instance, none of the themes identified by the text analysis tool deal with issues that focus on the second semantic category that encompasses hardware that contains or is controlled by software. However, the **DRB** concept occurs pervasively in all 2245 text blocks. As do other concepts in all three projects, it also co-occurs frequently with concepts in more than one semantic category.

Semantic Categories	Number of Text Blocks Where Theme-Naming-Concepts Occur
Information Manipulation, User Interface, and other Usability Factors	Displays (376), user (47), mission (displayed) (333), shows (90) (display), data (110)
Hardware System or Modules containing or controlled by information technology or software	None
Context of Use (Mission, Exercise, Training, User)	User (73), mission (333)
Testing and Maintenance, Configuration Management	DRB (2445) (Data Review Board), agreed (42), issue (304), impact (47), shows (90), data (110)
Software, Software System, Data, Data Standards	SRBD.App.C (70), mission (333) (data), shows (90) (doc) (requirement), data (110)
Systems and Software Engineering	DRB (2445) (Data Review Board)

Figure 19: Semantic Categories in Project 3

Project 3 has fewer themes than do the other two projects (Figure 20). Conceptual content is evenly distributed among all themes and is particularly dense in maintenance and testing themes such as **DRB** and **issue**, as well as in user interface- and usability-related themes such as **displays** and **mission** (see Figure 21).²⁰ Although data review boards (DRBs) function in the other two projects, they were not mentioned frequently enough in the other two projects' PCRs to emerge from the automated text analysis as themes or concepts.

²⁰ The **DRB** and **issue** theme names that are visible in Figure 20 are difficult to see in Figure 21 because of the density of the concept names.

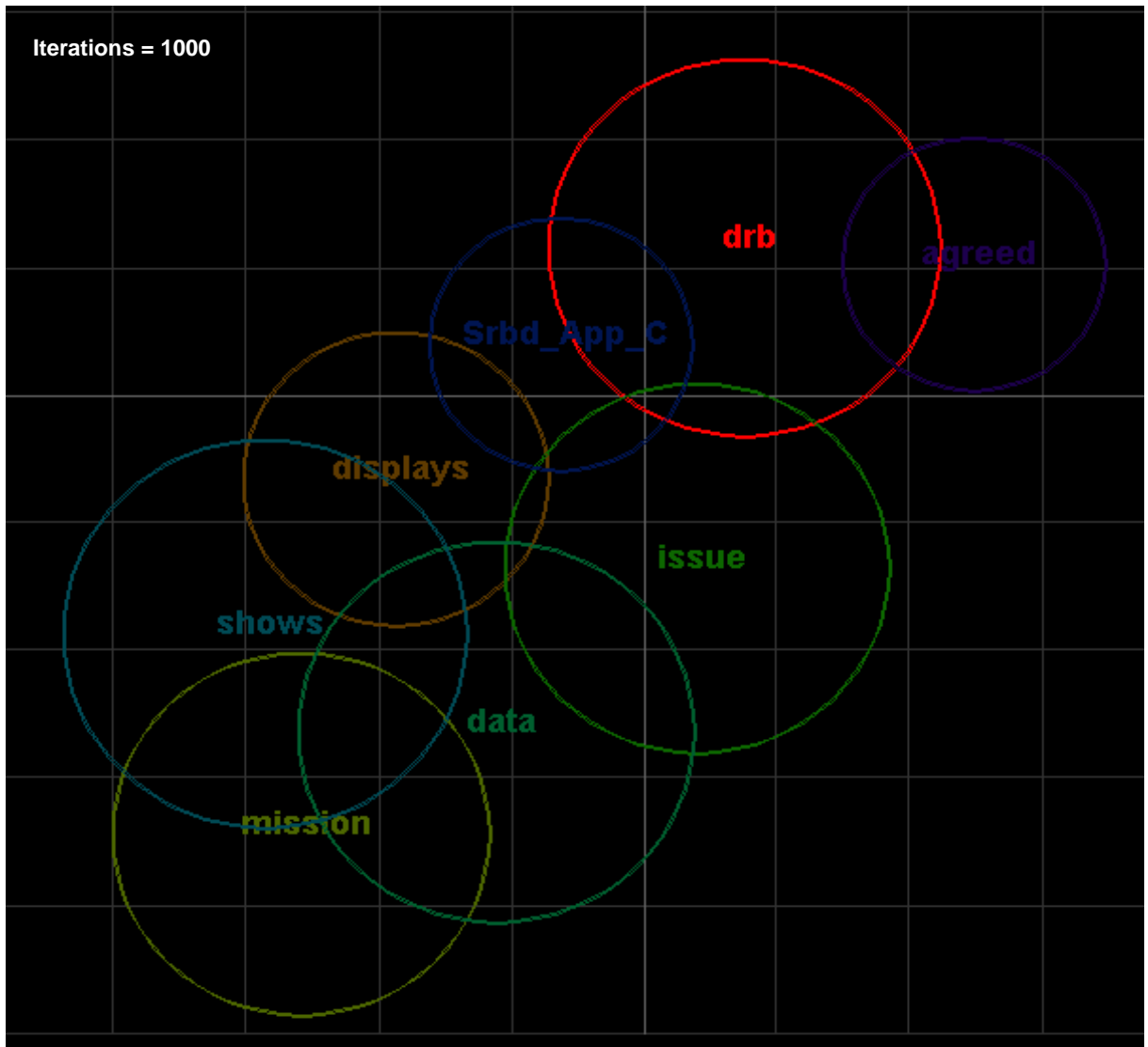


Figure 20: Project 3 PCRs Concept Map Showing Themes Only

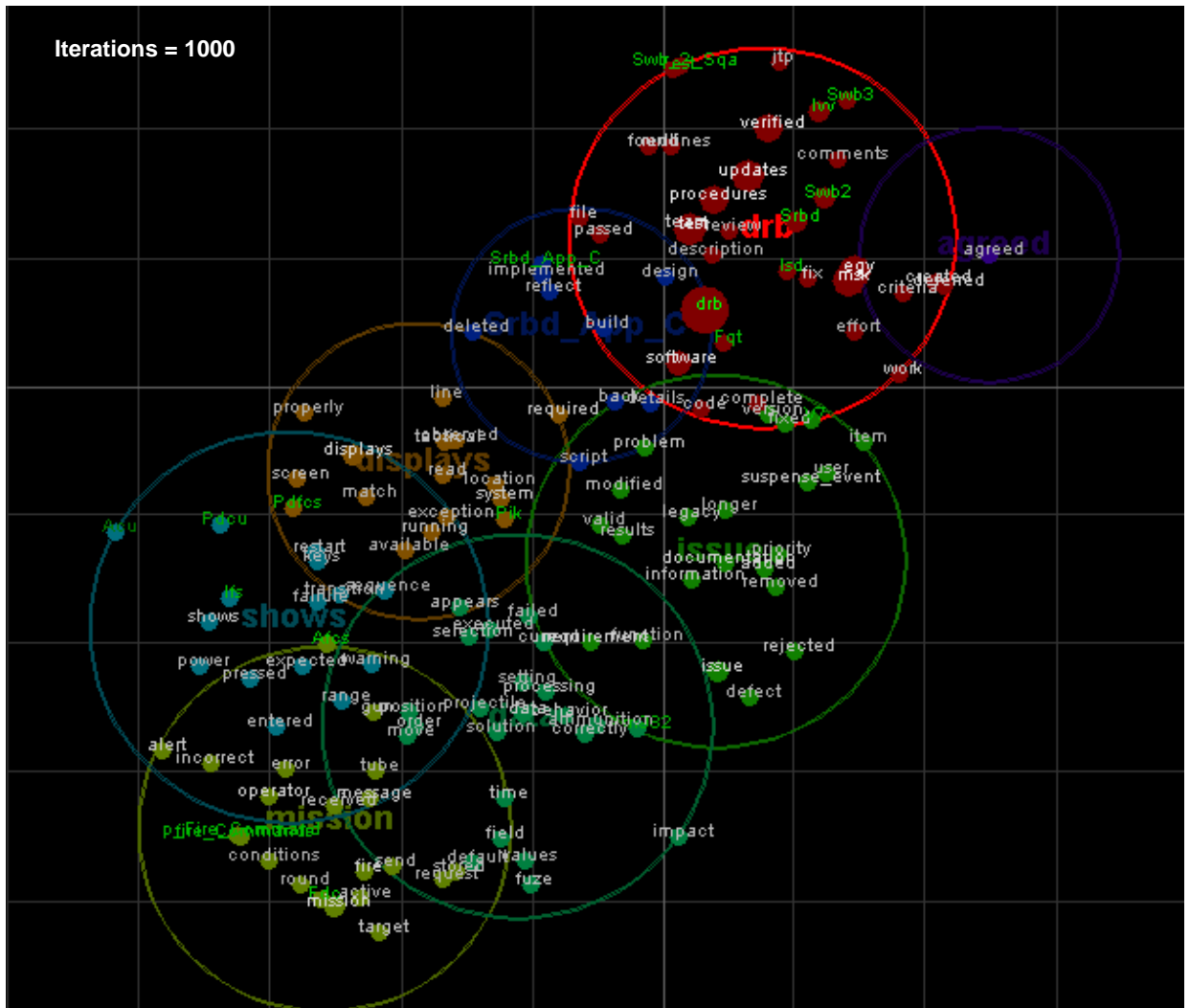


Figure 21: Project 3 PCR's Concept Map Showing Themes and Concepts

The makeup of the DRBs is the same in all three projects, consisting of technical leads (e.g., designers), project managers, and users or their representatives. The DRBs evaluate problems identified in the PCRs and decide what to do about them (e.g., prioritize, make changes, assign rework, defer the PCR, or reject it). The automated text analysis identified **DRB** as a concept and a theme in Project 3 because of its more frequent and interconnected usage of the term “DRB,” and its synonym list. Project 3 PCR's thus provides better insight into what the DRB does in its role in the PCR disposition process.

Figure 22 focuses on the most frequent concepts that populate the themes identified in Project 3 PCR's. These concepts, all of which are interlinked across themes, are most prominently linked through the **DRB** concept. These linkages begin to depict an emerging structural model of the Project 3 contents of PCR's. Moreover, the concepts in the **DRB** theme are the most frequent ones in the Project 3 PCR's.

A first step is to interpret the references of the different concepts in the work of the **DRB**. These interpretations are based on the way the terms corresponding to concepts are actually used in the

PCRs. For example, concepts in the **DRB** theme can be interpreted as follows. The DRB refers to **created** PCRs and calls upon one **team** or another in doing its **work**. In doing its **work**, it has **criteria** for deciding which PCRs should be **deferred**, whether a **fix** or **updates** to **software** or **code** are needed or already **verified**, whether a **test** or **IVV** (independent verification and validation) is **passed** or **complete** and/or in accord with **procedures**. Sometimes this involves information that is **found** in a **file** concerning **SWB_2**, **SWB 2_SQA** or **SWB_3** or in a **file** containing an **ISD** (interface specification document), which the **DRB updates** and **redlines**.

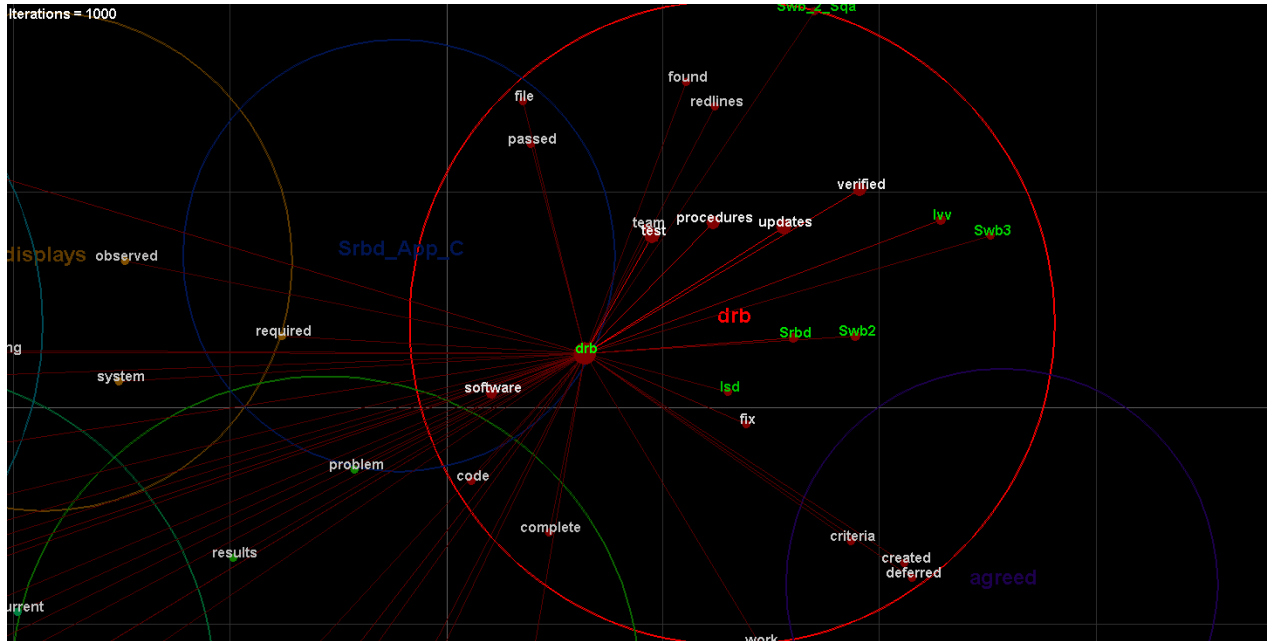


Figure 22: Most Frequent Concepts Populating the Theme **DRB** in Project 3 PCRs

Figure 23 shows how concepts in the theme **DRB** interrelate with concepts in other themes. These latter concepts in other themes tend to be less frequent than those in the **DRB** theme, but they still appear quite frequently (see Figure 24). For example, **criteria** for **created** PCRs that are **deferred** have to be **agreed** upon (since these concepts are located in the overlap of the **DRB** and **agreed** themes). With respect to the themes **issue** and **SRBD_APP_C** (which stands for Software Requirements Baseline Document Appendix C), the PCRs refer to the **DRB** as having **added** or **rejected** a **problem** or **issue** based on **results** of a **test** and informed by **documentation**, by a **requirement** or by what is **required** by the **SRBD_APP_C**.

Ranked Concept List

Concept	Absolute Count	Relative Count	
drb	2445	100%	
updates	1239	50.6%	
test	1137	46.5%	
verified	988	40.4%	
procedures	905	37%	
software	572	23.3%	
Ivv	527	21.5%	
Srbd	472	19.3%	
displays	376	15.3%	
mission	333	13.6%	
Swb2	331	13.5%	
issue	304	12.4%	
operator	239	9.7%	
deferred	237	9.6%	
fix	233	9.5%	
requirement	228	9.3%	
documentation	218	8.9%	
fire	200	8.1%	
Swb3	179	7.3%	
created	170	6.9%	
suspense_event	165	6.7%	
added	161	6.5%	
file	161	6.5%	
message	155	6.3%	
required	152	6.2%	
work	140	5.7%	
AfcS	134	5.4%	
system	126	5.1%	
team	118	4.8%	
passed	111	4.5%	
data	110	4.4%	
alert	107	4.3%	
rejected	107	4.3%	
Fdc	107	4.3%	
behavior	107	4.3%	
Isd	106	4.3%	
criteria	106	4.3%	
processing	106	4.3%	
code	101	4.1%	
problem	101	4.1%	
time	100	4%	
found	96	3.9%	

Figure 24: Ranked Concept List for Project 3 PCRs

Figure 25 shows concepts in the **displays**, **shows**, **data**, and **mission** themes that can be interpreted to describe the specific factors underlying usability-related **issues** that the DRB addresses. Many of these issues revolve around an **operator** interpreting **displays** of **data** and **messages**, especially an **alert** or **warning** (not shown) that is **received** on the **screen** in order to enter a **fire mission** to be sent to the **FDC** (Fire Direction Center) or **AFCS** (Automatic Fire Control System). Sometimes an **alert** is displayed inadequately or is misleading, or other **software behavior** is manifest on **screen displays** but is not documented or is inconsistent with **requirements**.

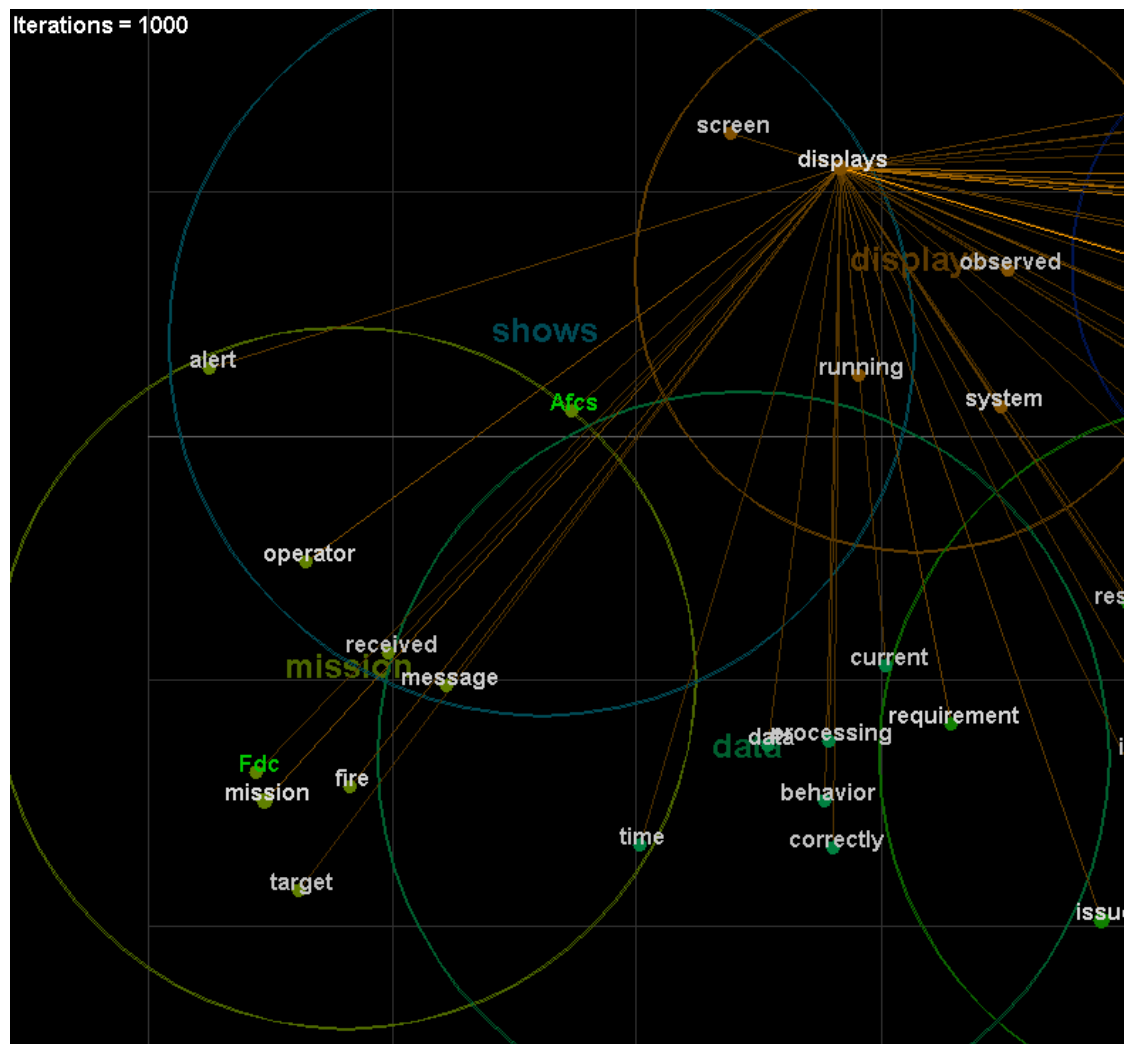


Figure 25: **Displays, Shows, Mission and Data Themes in Project 3 PCRs**

4.7 SYNOPSIS AND FUTURE OPPORTUNITIES

The semi-automated content analysis identified recurring usability-related issues that had not been fully recognized on a case-by-case basis. All of them were found and corrected prior to release; however, such issues may be mitigated or avoided by process improvements resulting from this and future analyses done by the SEC.

Usability issues were fairly frequent and significant across all three projects, although they can be characterized differentially. Analyses thus far have identified several kinds of usability-related issues, including

- identification and characterization of data entry errors and providing appropriate feedback
- proper layout and operability of soft buttons on a small PDA screen when reusing software that is adapted from a system using a desktop computer with larger screen and keyboard²¹
- display of inadequate and misleading alerts or warnings
- other software behavior manifested on screen displays that is not documented or is inconsistent with requirements.

These and other problems and issues can be elaborated further. For example, future analyses might generate results that fit scenarios based on those described by Bass and his colleagues [Bass 2003b]. Such categories could be divided into sub-categories of usability and other quality attributes such as reusability, modifiability, sustainability, and interoperability. Further analyses of current and additional documentation and other textual and quantified data provided by the SEC may help further refine the usability analysis described in this report. Further analyses of other quality attributes, their kinds, characterizations, and tradeoffs among them might also prove to be useful. Similarly, some usability issues are tightly coupled to software architectures. Others are separable yet still relevant to higher level operational and system architectural considerations, while others are not.

The potential value of capturing the conceptual space of quality attributes in the PCRs has implications for requirements development in general. Specification of quality attributes in requirements and other non-architectural documentation provide a basis for validation of eventual products, whether they are related to systems, subsystems, and modules or systems of systems. It may provide a better basis for validating the respective architectures of these products as well as building both system and operational architectures. Moreover, validating the architectures may be quite useful in validating the products themselves.

Regardless of the topics, follow-on analyses should be based on semi-automated content analyses, which would be more fully elaborated in collaboration with SEC members and their key stakeholders. Continued collaboration with the SEC is important because they are familiar with the contexts of use addressed by the systems sustained and developed in the three projects. In addition, interaction with program managers, users, and technical people would be extremely valuable. A quantified range of operational, system, or software scenarios and response measures acceptable for each type of usability and other quality attributes could then be identified and specified.

The results described in this report demonstrate what can be accomplished by semi-automated content analysis. It can facilitate the distillation and resolution of problems and issues into quality attributes. These quality attributes can be categorized, subcategorized, and characterized in scenarios where the range of acceptable or desired operational, system, or software responses can be

²¹ Architectural solutions to such usability-reusability-modifiability tradeoffs may be possible (e.g., by separating a system's user interface from its functionality to support iterative design and reusability).

quantified and used as a basis for better software engineering measurement and analysis. Seeding additional concepts based on practitioner-collaborators' in-depth knowledge may be especially useful. Such values would not simply be terms such as usability, reusability, modifiability, sustainability, or interoperability. They could be instantiations of one or more scenario schemas subject to tradeoff analysis in terms of which quality attribute might take precedence in a given context.

Such schemas and tradeoff analyses also could benefit by the iterative creation of a semantic formalization or ontology [Masolo 2003]. A formal ontology possibly could provide the basis for much better computational or automated support for specification of quality attribute requirements as well as improved PCR processes and documentation.²²

Use of quality attributes in PCRs, for example, could begin by selecting a quality attribute as values of a field like `Quality_Characteristic_Affected` that earlier was removed from use in the Project 2 PCRs. Based on analysis of the PCRs and interaction with the project personnel, the authors recommend that this field should be re-considered for future use in the PCRs. Its reintroduction should be accompanied with more adequate support for selecting appropriate quality attributes and articulating tradeoffs among multiple quality attributes that may be applicable to a given PCR.

A formal ontology could provide the basis for a computational environment that would support specifying quality attributes into objectives, scenarios, measureable thresholds, and desirable outcomes that a responses should achieve. The environment could support linking or including quality attributes in requirements specifications or PCRs in a collaborative fashion and could also facilitate interactions at a distance with users concerning usefulness and usability.

Finally, dissemination of analysis results and suggestions for computational support might be of use to the Six Sigma groups established to improve PCRs and requirements specification at the SEC.

²² There has been an increasing focus on information content in recent years for building complex information and communications systems, with explicit conceptual models of the environments in which the systems operate, the organizations in which they are used, and the data and knowledge that they process. Ontology is best understood as a general theory of the types of entities and relations that make up a particular business, military or other domain and the systems that operate within it [adapted from Guarino 2005].

5 Summary and Conclusions

The SEC already has mature processes, process assets, and robust delivered systems. Applying semi-automated content analysis in a proactive approach to causal analysis may further its efforts at continuous process improvement.

Most of the results found in this study are consistent with what the practitioner-collaborators already know about their systems. Many of the semantic classifications, including some that map well to quality attributes, are in fact well understood by the software engineering center personnel (e.g., system reliability, dependability, and accuracy). Such issues are already incorporated in the performance measures and acceptance criteria that are used at the SEC. However, analysis of the problem and change requests also identified concepts and themes that appear to map well to usability issues that may not be fully anticipated in the requirements specifications or test procedures.

The prevalence in the SEC's PCRs of issues related to quality attributes such as usability, readability, informability, and knowability suggests that consideration of these attributes earlier in the life cycle, even before creating an architecture, may lead to significant improvement. This is true whether key design considerations are determined in the requirement specifications that the SEC receives or other constraints are introduced elsewhere. Quality attribute considerations are also worth capturing in architecture documentation, not just for software but for system and operational architectures as well. The latter would be derived in part from the quality attributes that JCIDS capability documentation calls KPPs.

There also may be opportunities for improvement of the SEC's verification and validation processes. For example, one project stopped using a field on its PCRs that was meant to capture quality-attribute-related problems. It may be wise to reconsider that decision, accompanied by more detailed processes, training, and measurement definitions. Similarly, whether for immediate corrective action or future releases, there may be opportunities to improve the SEC's delivered systems, system requirements, and their traceability with desired force capabilities.

It remains to be seen if further causal analyses and resolution activities will identify actionable improvement plans with respect to usability. However, several things suggest that further efforts to improve requirements processes to address usability across the life cycle may be valuable. Along with published literature from warfighters and combat developers, these include the simple existence of usability problems as identified in the text analysis, as well as the SEC's existing processes to capture user perspectives through integrated product teams, operational scenarios, and the employment of recent combat developers and warfighters.

Maintainability and modifiability do not appear to be major problems at the SEC, and semantically related terms do not appear as concepts in the text analysis. The SEC has existing processes, software design criteria, and coding conventions that are followed to correct offending code when it is found during routine maintenance procedures. Still, the terminology in use differs among the three projects as well as in the three kinds of data sources used for the semi-automated content analysis. Further analysis may yet uncover opportunities for improvement in this area.

5.1 IMPLICATIONS AND LIMITATIONS

The semi-automated content analysis methods themselves and the results derived by using them usually do not provide the basis for immediately actionable solutions, although the results sometimes can be used to guide specific corrective actions. Rather, their major contribution is in helping developers, maintainers, and other affected stakeholders better understand problems that then can be addressed using standard engineering methods. For example, the content analysis results may help identify new opportunities for improvement in existing processes or identify issues for escalation beyond an organization's current scope of control. These methods and tools provide unique opportunities for proactive causal analysis by reviewing voluminous amounts of data to uncover recurring patterns that may have been missed in case by case adjudication of the PCRs.

The extent to which usability issues are considered before testing and maintenance needs to be investigated further. Operational scenarios and other documents generated in the three projects that do consider and operationalize usability issues may provide a sufficient basis for maintenance or sustainment work prior to testing. Processes also exist in the SEC to capture warfighter perspectives. Individuals with recent field experience are employed in systems engineering roles. PMO representatives, and to a lesser extent, combatants and their representatives, are queried through the auspices of an existing integrated product team (IPT).

Moreover, the results described in Section 4 remain provisional. They should not be over interpreted as being either conclusive or broadly generalizable elsewhere. The analysis thus far has demonstrated that semi-automated content analysis can quickly identify recurring patterns of related text about certain topics that might not be considered otherwise. The PCRs describe problems related to user interface and usability. However, further in-depth causal analysis by domain experts is necessary to determine whether or not these problems fall into common categories that could have been anticipated or prevented.

The results presented in this report are from only one Army SEC, with a perspective that is unique compared to other sites that are providing similar documents for content analysis. These other sites are Program Executive Offices (PEOs) that oversee multiple acquisition projects rather than development or maintenance projects.

Regardless, operational capabilities remain a source of concern in Army maintenance organizations just as they do in the Program Management Offices (PMOs) overseen by the PEOs. Some quality attributes, and for the purposes of this study usability in particular, are not well understood conceptually; hence, they often are not documented adequately or explicitly. Capable projects and organizations sometimes struggle mightily with them. For example, almost half of the respondents from high maturity organizations in a recent survey said that they used quality attribute measures of any kind only occasionally at best [Goldenson 2007]. When quality attributes are not considered explicitly as operational or system requirements, acceptance criteria and other performance measures will focus heavily or exclusively on system functionality.

The meaning and utility of quality attributes must be made clear in practitioners' own contexts if such concepts are ever to be applied effectively. This includes the operational contexts for which capabilities are defined as well as the system and software contexts for which requirement specifications are defined. Richer specification of quality attributes in both contexts, especially with respect to usability, will enable better traceability between customer and systems requirements

that is so crucial for validation. Incorporating semi-automated content analysis methods into an organization's ongoing causal analysis and resolution processes may provide a basis for establishing such traceability (see Section 5.2).

Semi-automated content analyses also may lead to improvements in the policy documents and process and quality models that are meant to guide practitioners. Results from analyses that focus on similar problems across particular practical contexts may suggest opportunities for improvement in the models and frameworks themselves. Opportunities for improvement can be facilitated if the same kinds of analytic methods are applied directly to the texts of the documented models and policies themselves. An example of such use in analyzing the full text of CMMI-ACQ may be found in a conference presentation by the present authors usually [Monarch 2008].

5.2 CONCLUSIONS AND FUTURE WORK

The research described and initial results presented in this technical report provide proof of concept that semi-automated content analysis can help practitioners identify opportunities for improvement in their products and work processes that might otherwise go unrecognized. By extension, they suggest that semi-automated content analysis methods can be used to improve our understanding in this and other important areas of empirical software and systems engineering. However, much more work remains to be done. Many more sites need be analyzed and joined with other measurement approaches to make more definitive claims about the state of requirements engineering practice in the Army or elsewhere. The same is so for other areas of research that may benefit by advances in semi-automated content analysis.

Plans are underway to continue the analyses at the Army software engineering center whose work is described here. Our practitioner-collaborators there have identified additional documents that can be analyzed, including design documents, training documents, operational scenarios, and field reports. In addition, they have suggested several opportunities for further collaboration. Staff members have downloaded some of the analytic software used for this report, and discussions have begun about ways to incorporate content analysis methods into existing causal analyses processes and on-going Six Sigma studies at the SEC. These may facilitate analyses of the impacts on project performance and product quality of future process changes, the establishment of new working relationships or the introduction of new technology.

Regularly doing content analysis may identify changes in the problem space earlier. Patterns of use found in analyses of existing data can be used as a basis for improving new releases and new sustainment projects. They also may suggest useful changes to forms and related processes to better track changing requirements. It also is possible to join qualitative data into a common measurement repository database linked to an organization's process asset library.

Additional plans and analyses are ongoing with other Army and joint force sites. Organizations that have participated in extensive discussions and made documentation available for analysis include the Joint Program Executive Office Chemical-Biological Defense (JPEO-CBD), Joint Requirements Office Chemical-Biological Radiological Nuclear Defense (JRO CBRND), Army Program Executive Office (PEO) Soldier and Project Manager (PM) Soldier Warrior, Army PEO Aviation, and Army PEO Command, Control and Communications – Tactical (C3T).

Work underway elsewhere is aimed at better aligning customer-desired capabilities and quality attributes with derived requirements in legacy systems. Documents made available for analysis include Initial Capability Documents (ICDs); Capability Development Documents (CDDs); Capability Production Documents (CPDs); and Operational Requirements Documentation (ORDs). Derived Requirements Specifications include Implementation Plans (IPs); Information Support Plans (ISPs); Software Problem Reports (SPRs); and Problem and Change Reports (PCRs). Related documentation and records also exist that can and should be traceable to the capability documents. These include Military Operational Concepts and Doctrine; information captured in vetting of capability documents, architectural and design documents; testing scripts; other intermediate outcomes and final results; problem reports and change requests from testing, training, and the field; and maintenance and sustainment outcomes.

Most organizations do not phrase quality attributes in clearly defined scenarios and quantified terms, so they typically find the kinds of defects that they anticipate. Other collaborations similar to the one described in this report may lead to better training mechanisms, including more formalized hands-on workshops.

In principle, semi-automated content analyses can be done at any aggregated unit of analysis. Detailed analyses need not be limited to individual projects. Text from larger organizations can be analyzed together to identify common, shared problems to provide better decision support for portfolio management. The same is so among components of a system of systems. Content can focus on commonality as well as individual cases. Text analyzed can be aggregated over product components, component interoperability, requirements statements, test procedures, or problem and change reports from separate projects, organizations, larger enterprises, or systems of systems. Serious consideration is being given to doing further analyses of CMMI model structure and content, as well as other important policy documents, process, and quality models.

Another promising approach could use semi-automated content analysis in concert with collaborative software [Boehm 1998]. Doing so could be particularly valuable for eliciting additional information from large numbers of stakeholders, especially those who are not co-located geographically. Collaborative software works essentially by increasing participation in virtual group discussions where text is entered, reviewed, and clarified by participants online. Mechanisms exist to encourage open participation, which can capture more fully explicated and complete textual records for analysis. Such tools have been used elsewhere in the Army and with ships at sea [Army 2003]. Typical applications in addition to requirements engineering include project planning and portfolio management. Collaborative software also has been used without the text analysis to analyze inspection productivity [van Genuchten 2001].²³

Opportunities for improvement and exemplary practices need to be better understood in the context of particular organizations before they can be generalized elsewhere. Semi-automated content analysis is a relatively inexpensive way of focusing attention on important concepts by analyzing documented discourse among various practitioner stakeholders in their own terms. Practitioners can see value in this way of proceeding because policies, processes, and quality models can be

²³ van Genuchten also used the same collaborative software in his unpublished work on software process appraisals, noted in footnote 27 on page 54.

better understood in their own context. The focus is on *how* things are done, not just what should be done.

Not only can the methodology be adopted by practitioners to improve their own organizational bodies of knowledge locally, analyses of this kind also may enable practitioners to have a greater hand in policy making and model construction. As more work of this kind is done, it is our hope that the results will be compared across organizations and collected into useful lessons learned repositories, and that they will influence the content and value to practitioners of future policies, process, and quality models.

In the end, our goal is to mature the semi-automated content analysis methods and procedures such that they can be used by software and systems engineering practitioners with minimal outside guidance. We hope that this report provides a viable beginning.

Appendix A: Further Background on Quality Attributes and Usability

JCIDS KEY PERFORMANCE PARAMETERS

Large amounts of heterogeneous information from multiple disparate capability stakeholders must be understood, coordinated, and synchronized across organizational and disciplinary boundaries to provide an adequate basis for capability development. JCIDS has provided policy and guidelines to identify and structure this information and to facilitate its flow via various capability documents and processes [CJCSI 2007, CJCSM 2007]. The JCIDS policy addresses quality attributes in capability development mainly through Key Performance Parameters (KPPs) and Key System Attributes (KSAs). The KPPs are broader categories, and the KSAs are finer grained categories that help define the KPPs. Measured attributes are “value determiners” that help determine the values of the KPPs and KSAs, as seen in Enclosure B of the JCIDS manual [CJCSM 2007]. There also are attributes that fall outside of or are not emphasized in the JCIDS specification of KPPs, KSAs, and value determiners. Usability, or what JCIDS calls Human Systems Integration, is an important example as seen Appendix A, Enclosure F, and Glossary GL 9 of the JCIDS manual.

Examples of quality attributes in JCIDS terms include

- **survivability** KPPs like speed, maneuverability, detectability, and countermeasures reducing likelihood of being engaged by hostile fire
- **sustainment** KPPs such as materiel availability and its two supporting KSAs, materiel reliability and ownership cost
- **net-ready** KPPs like interoperability that are to be used in information support plans to identify support required from outside a program
- **KPPs covering characteristics of the future force**, including being knowledge empowered, networked, interoperable, expeditionary, adaptable/tailorable, enduring/persistent, precise, fast, resilient, agile, lethal
- **information assurance** KPPs that protect availability, integrity, authentication, confidentiality, and non-repudiation

Suitability is used in JCIDS as a “higher order” KPP. It is defined as:

The degree to which a system can be placed and sustained satisfactorily in field use with consideration given to availability, compatibility, transportability, interoperability, reliability, wartime usage rates, maintainability, environmental, safety, and occupational health, human factors, habitability, manpower, logistics, supportability, logistics supportability, natural environment effects and impacts, documentation, and training.

Attributes in this definition such as human factors, habitability, and wartime usage rates are not defined elsewhere in JCIDS. However, the manner in which usability or any other quality attribute is being handled in JCIDS ICDs, CDDs, and CPDs can be investigated using content analysis. As noted in Section 1.1 and described more fully in Section 4, usability-related issues have in fact been recognized in content analyses applied at the SEC analyzed in this report and elsewhere.

OTHER QUALITY ATTRIBUTE SCHEMA

As noted earlier in Section 2.1, several existing standards have addressed quality attributes. Figure 26 summarizes the classification schema used in ISO/IEC 9126-1, which is the software product quality standard produced jointly by the International Organization for Standards (ISO) and the International Electrotechnical Commission (IEC) [ISO/IEC 2001]. Six high-level characteristics are broken down into several related sub-characteristics.

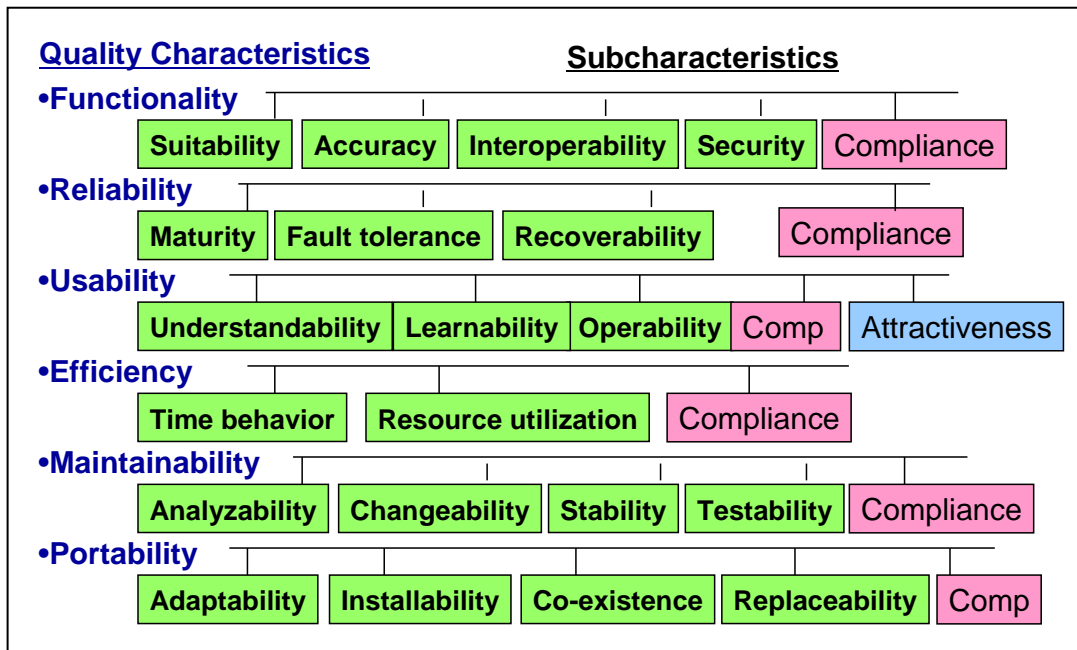


Figure 26: ISO/IEC 9126-1: Software Product Quality

Note that the categories are similar to the Key Performance Parameters (KPPs) and Key System Attributes (KSAs) that are called out in the Joint Capabilities Integration and Development System (JCIDS) policy (as discussed in the previous section of this Appendix). However, they are not the same.

Similarly, the categories used by leading software architects differ in subtle and not so subtle ways (see Figure 27). For example, the top level system quality attribute categories used in Bass, Clements, and Kazman's highly regarded work [Bass 2003b] are availability, modifiability, performance, security, testability and usability. Availability for the architects overlaps significantly with reliability for the ISO/IEC standard. Modifiability and testability both are largely subsumed under maintainability. Security is a sub-characteristic of functionality for the ISO/IEC authors, and performance is orthogonal to the quality attributes in 9126-1. Only the term "usability" is used in a somewhat more directly comparable manner in both sources.

None of the three sources (JCIDS, ISO/IEC 9126-1 1999, or Bass 2001, Chapter 4) is more correct or accurate than the others. While one can hope for better harmonization among them and others as more is learned over time, all were created for different purposes. What is important is that they can help focus system architects as well as capabilities and requirements developers on important problem areas that frequently arise elsewhere. Much like the goal, question, metric (GQM) paradigm [Mashiko 1997, Goethert 2007] the trick then is to clarify, refine, and prioritize

the more general quality attribute categories into finer grained, measureable terms that are pertinent for use under particular operational circumstances or incorporation into a particular system. As more organizations do such refinement and their experiences are incorporated into existing quality standards and frameworks, the standards and frameworks themselves may become more easily accessible and useful to software and systems engineering practitioners.

	Source of Stimulus	Stimulus	Artifact	Environment	Response	Response Measure
Availability (Reliability)	External to System	Unanticipated Message	Process	Normal Operation	Inform Operator	No Downtime
Modifiability (Maintainability)	Developer	Change UI	Code	Design Time	Modification & No side effects	In three hours
Performance (No)	Users	Initiate Transactions	System	Normal operations	Transactions processed	Average latency of two seconds
Security (Functionality)	Correctly Identified Individual	Tries to modify Information	Data within system	Normal operation	System maintains audit trail	Correct data restored within a day
Testability (Maintainability)	Unit Tester	Performs Unit Test	System component	Component completion	Component behavior observed	85% path coverage achieved within 3 hours
Usability (Usability)	Users	Minimize impact of errors	System	Runtime	Cancel current operation	Cancellation takes less than one second

*SEI architecture usage: ISO/IEC 9126-1 terminology in parentheses

Derived from *Software Architecture in Practice*, Chapter 4 [Bass 2003b].

Figure 27: SEI Software Architecture Quality Attribute Scenarios

USABILITY

Usability has been a particularly poorly understood concept in software and systems engineering. More and better collaboration with experts in human-computer interaction is needed. As is true with respect to quality attributes in general, aspects of usability are treated differently by different sources. ISO/IEC 9126-1 (1999) partitions usability into sub-characteristics of “understandability,” “learnability,” and “operability.” As Bass, *et al.* describe on pages 90-91 of their book [Bass 2003b], “learning system features” overlaps with understandability and learnability in 9126-1, while both “using a system efficiently” and “minimizing the impact of errors” are comparable to operability for the ISO/IEC authors. “Increasing confidence and satisfaction” is a fifth sub-area of usability for Bass and his colleagues; it overlaps to some extent with the “attractiveness” usability subcategory that is currently being considered for use in ISO/IEC 25000, which is slated to replace 9126-1 and related existing standards. Bass, *et al.* emphasize on page 92 of their book that, “the usability features that are the most difficult to achieve (and, in particular, the most difficult to add on after the system has been built) turn out to be *precisely* those that [also] are architectural.”

Other notable sources include Krippendorff, whose discussion of “disruption” is quite helpful. As seen in Figure 28, system use can be disrupted by various sorts of system or user interface errors and user slip-ups or mistakes that interfere with both routine and non-routine tasks. There is much anecdotal evidence that sometimes the situation can be bad enough from the users’ perspective that they cease using important functions or the system altogether.

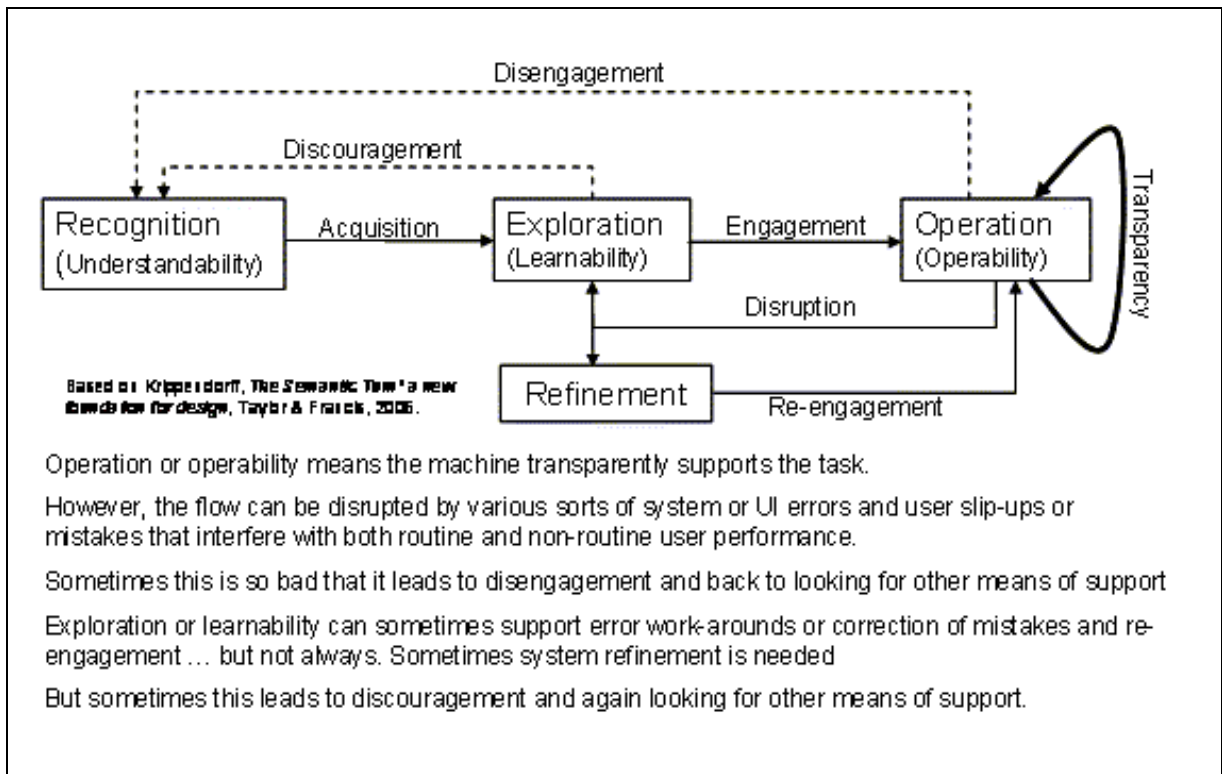


Figure 28: Understanding a Key Quality Attribute – Krippendorff's Usability

Other important aspects of usability include the following:

- Traversal – Can the user find what he or she needs when it is needed? Hierarchical menus or other structured data hiding techniques come to mind here, as do Edward Tufte's notions of clutter and "chart junk" [Shneiderman 2004, Tufte 1983, Tufte 1997].
- Clarity – Can the user easily interpret the visual displays, layout, and audio queues? [Shneiderman 2004, Tufte 1983, Tufte 1997].
- Notification – Are warnings and alerts presented on a timely basis, clearly, and without unduly interfering with current activities? [Bass 2003b]
- Returning to previous state – Can the user easily recover from errors, check on progress, or multitask in other ways? [Bass 2003b]

Appendix B: Further Background on Semi–Automated Content Analysis

TRADITIONAL CONTENT ANALYSIS

Content Analysis has been a standard methodology in the behavioral sciences for many years [Krippendorff 2004, Neuendorf 2001, Berelson 1952, Weber 1990]. It has been used for studying the content of printed documents and other communications by using systematic, replicable techniques for compressing many words of text into fewer content categories based on explicit rules of coding. For example, it was used in World War II to predict the bombing of London by the Germans by analyzing the content of Joseph Goebbels’ speeches [Krippendorff 1980].

Content analysis of free form text sometimes is used in public opinion and other survey research to estimate specific percentages and other population parameters. However, its major contribution is to better understand the nature of previously unstructured problem areas and clarify as yet poorly understood concepts. Similar to focus groups in modern survey research, content analysis results more typically are used to clarify ideas and suggest useful categories with clear operational definitions for measures that can be used in subsequent analyses. In that sense, content analysis can serve as a forensic tool that can provide clues and suggest where refinements may be needed during further causal analysis and resolution.²⁴

The problem with traditional content analysis is that it is very time consuming and difficult to do. In fact, that is why survey researchers since the 1950s have relied much more heavily on closed-ended questions that require choices among pre-defined response categories. Open-ended free form responses in people’s own words are much harder to analyze. Not only must the analysts create well-defined categories, they also must code the open-ended text consistently and reliably. There are well-defined statistical methods to check for what is called inter-coder reliability, but the process can be extremely time consuming and error prone [Krippendorff 2004, Banerjee 1999]. The semi-automated tools and techniques used in the present study reduce time consumption and difficulty considerably and reduce the need for inter-coder reliability since the algorithms used by the automated content analysis tool are not applied subjectively. Errors are reduced and huge amounts of data that was not being analyzed now can be.

AUTOMATED TEXT ANALYSIS

Automated text analysis tools have existed since the 1960s.²⁵ They rely on computational and linguistic algorithms, which are based on underlying mathematics similar to those used for pattern

²⁴ Note that content analysis, whether using manual or automated methods, has to address synonymy (different words having similar meaning) and polysemy (the same word having several meanings) in order to provide accurate counts. People typically do not think in the same contextual terms, particularly when considering poorly understood or unfamiliar topics. The tools and techniques being used and developed in this study are very sensitive to these problems and mitigate them considerably.

²⁵ An early program described in [Stone 1966] still is being used by some quantitatively oriented behavioral scientists; see <http://www.wjh.harvard.edu/~inquirer/> for more detail. Brief descriptions and links to other, more recent examples can be found at http://en.wikipedia.org/wiki/Text_mining#Software_and_applications.

recognition, data reduction of quantitative measures, and dimensional analyses such as factor analysis. Various combinations of lexical and natural language techniques are used to identify and thematically categorize co-occurrence of similar words and phrases. Some also provide functionality for joining those categorizations for analyses with other existing quantitative data [Galt 2008, Coulter 1998]. Similar tools are used for various internet and other data mining purposes.

Examples where text analysis has been used include studies of thematic differences between software practitioners and published research with respect to measurement processes and related issues [Monarch 2005]; process appraisal methods [Dunaway 1998]; appraisal findings²⁶; thematic changes over time in published software engineering research [Coulter 1998]; and risk information analysis [Monarch 1995]. In addition, text analyses have been used to derive findings from appraisal interviews.²⁷ Ongoing work and similar analyses have been done elsewhere at U. S. Army sites. Other notable work has been done in library science and in medical research to identify promising treatment modalities.

Of course, automated text analysis results still must be interpreted by humans with appropriate domain expertise. Their value is in identifying underlying patterns that would be difficult if not impossible to discover with manual methods. As seen more fully in Section 3 and later in this appendix, the tools also narrow the search space and enable smart searches that help analysts corroborate and clarify the sometimes unanticipated patterns identified by the automated tools.

THE NEED FOR BETTER KNOWLEDGE MANAGEMENT

As noted in Section 2, the traceability of Army capabilities documentation through system requirements is very difficult to manage. The same is true in many complex commercial and industrial settings where varying stakeholder perspectives must be considered. That is due in large part to differences in perspectives among key stakeholders, compounded by differing organizational responsibilities, along with incompletely understood and changing operational and system requirements. Online keyword search is a major improvement over traditional card catalogs; however, the lack of a common language semantics that is shared by all relevant stakeholders remains a much bigger problem. Many words take on different meanings in different contexts, for example “change” can refer to a major requirements change or a simple display change. Similarly, the same term often has different meanings: “issue” can refer to a problem or a means to provide something. Methods of conceptual indexing [Woods 1997] and conceptual search [Guarino 1999] exist, but they are not currently well-integrated into the analysis and evaluation work involved in requirements specification and traceability. Even relatively effective document management systems are limited by inadequate conceptualization [Mika 2003]. And quality attributes are loosely specified, if they appear at all, in requirements specifications [Ozkaya 2008].

²⁶ The report “CMM®-Based Software Process Improvement: Common Organization Issues Preventing KPA Satisfaction” is available for registered users at <https://seir.sei.cmu.edu/seir/domains/CMMspi/Benefit/ica/frmset.welcome.html>.

²⁷ Unpublished work in the Netherlands by Michiel van Genuchten in the late 1990s.

OUR APPROACH

Semi-automated content analysis combines automated text analysis with semantic classification, inference, and validation in collaboration with expert practitioners. The text analysis described in this technical report was done using a tool called Leximancer that was developed initially at the University of Queensland in Australia.²⁸

Text Analysis

Leximancer has excellent thematic analysis capabilities and capabilities for handling synonymy. It works through a progression from many analysis passes through the full text to extraction of concepts by collecting synonymous terms in synonym sets, and then to clustering the concepts in themes. The concepts essentially are automatically generated synonym lists of strongly related co-occurring terms in automatically determined blocks of text. The term most strongly related to the other terms in the synonym set becomes the name of the concept. The themes are collections of co-occurring concepts. They are based on strength of inter-relatedness and frequency of occurrence, and they are automatically named by selection of the concept most strongly related to the other concepts in the theme.²⁹

The tool starts by selecting a ranked list of important lexical terms on the basis of word frequency and co-occurrence of usage in the full body of text that is examined. These terms then seed a bootstrapping thesaurus builder, which learns a set of classifiers from the text by iteratively extending the seed word definitions. The resulting weighted term classifiers are then referred to as “concepts.” The text then is classified using these concepts at a high resolution, normally every three sentences. Doing so produces a concept index for the text and a concept co-occurrence matrix. By calculating the relative co-occurrence frequencies of the concepts, an asymmetric co-occurrence matrix is obtained.

The co-occurrence matrix then is used to produce a two-dimensional concept map via an emergent clustering algorithm. The connectedness of each concept in this semantic network is employed to generate a third hierarchical dimension, which displays the more general parent concepts at higher levels called “themes.” As seen in Sections 3 and 4, the themes are represented spatially as Venn diagrams. Each theme is shown as a circle, and its placement is based on closeness of meaning to the other themes. Concept placement is also based on closeness of meaning, and concepts can overlap theme boundaries, so the themes are not orthogonal. In the analyses performed for the SEC data, themes often overlap. Circle size is based on the placement of concepts clustered in a theme.

Semantic Analysis

The automatically generated themes and concept maps vary based on the level of abstraction chosen for a particular purpose. Which representation is best depends on the need for detailed nuance

²⁸ The product is described more fully at <http://www.leximancer.com/cms/>. As noted in this appendix, many such tools exist, and they have different strengths and weaknesses. The SEI does not rank or promote them in any way.

²⁹ More details about how Leximancer works and its results can be found in the article, “Evaluation of Unsupervised Semantic Mapping of Natural Language With Leximancer Concept Mapping” [Smith 2006].

versus broad generalization. Further semantic classification, inference, and validation must be done once the basic text analysis is complete. Semantic analysts must apply their background and contextual knowledge to interpret, classify, and refine the automatically generated themes and concept maps.

SEI analysts did the initial review of the conceptual mappings for this technical report. They searched through and read the text classified by concepts and themes to infer the existence or absence of quality attributes and other conceptual content. Further semantic analysis, clarification and validation then was done through face to face presentations and interviews with Army practitioner-collaborators who of course were more familiar with the documents that were analyzed and the organizational and product context in which the documents were used.³⁰

SEI analysts then did additional Leximancer analyses, the results of which can be seen in Section 4. Leximancer has less well-developed natural language processing capabilities than do some other content analysis tools; however, it has excellent capability for detecting similarity of meaning for generating synonym lists and concepts, for organizing concept co-occurrences for generating themes, and for indexing text blocks according to concepts. The latter allows the analyst to drill deeper and to do more focused searches through the automatically generated thematic and conceptual structure. Doing so helps the analyst to establish semantic categories that are supported by the textual evidence although not generated automatically. The process can and should continue iteratively to provide further corroboration and enhancement of the semantic interpretations. There is always a need for practitioners to discuss things using their own local terminology; however, semi-automated content analysis can help them tease out and share their common expertise in a common conceptual framework.

³⁰ Unlike traditional, manual content analysis, which uses inter-coder reliability methods to validate its classifications, the approach in this study emphasizes working with expert groups as is often done in root cause analyses.

References/Bibliography

[Abner 2007]

Abner, H. "The Army Requirements Process," AUSA Industry Day, Panel #4, Army Capabilities Integration Center (ARCIC), TRADOC, 6 June 2007.

[Army 2003]

Headquarters Department of the Army. *The Army Knowledge Management Implementation Plan* September 2003. Prepared by Office of the Army Chief Information Officer/G-6.
<http://usacac.army.mil/CAC2/BCKS/ArmyKMImplementationPlan.pdf>

[Barbacci 2003]

Mario R. Barbacci, Robert J. Ellison, Anthony J. Lattanze, Judith A. Stafford, Charles B. Weinstein, & William G. Wood. *Quality Attribute Workshops (QAWs), Third Edition*, CMU/SEI-2003-TR-016, ADA418428. Software Engineering Institute, Carnegie Mellon University, 2003.
<http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03tr016.pdf>

[Banerjee 1999]

M. Banerjee, M. Capozzoli, L. McSweeney, & D. Sinha. "Beyond Kappa: A Review of Interrater Agreement Measures." *The Canadian Journal of Statistics* 27, 1, 1999: 3-23.

[Bass 2001]

Len Bass, Bonnie E. John, & Jesse Kates. *Achieving Usability Through Software Architecture*, CMU/SEI-2001-TR-005, ADA393059. Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/pub/documents/01.reports/pdf/01tr005.pdf>

[Bass 2003a]

Len Bass & Bonnie John. "Linking Usability to Software Architecture Patterns Through General Scenarios." *The Journal of Systems and Software* 66, 2003: 187-197.

[Bass 2003b]

Len Bass, Paul Clements, & Rick Kazman. *Software Architecture in Practice, Second Edition*. Boston, MA: Addison-Wesley Publishers, 2003 (ISBN: 0321154959).
<http://www.sei.cmu.edu/books/>

[Berelson 1952]

Berelson, Bernard. *Content Analysis in Communication Research*. Glencoe, Ill: Free Press, 1952.

[Boehm 2007]

Barry Boehm. "Future Challenges and Rewards for Software Engineers." *Software Tech News* 10, 3 October 2007.

[Boehm 2000a]

Barry W. Boehm. "Project Termination Doesn't Equal Project Failure." *IEEE Computer* 33, 2000: 94-96.

[Boehm 2000b]

Barry W. Boehm, Dan Port, and Mohammed Al-Said. “Avoiding the Model Clash Spiderweb.” *Computer*, Nov, 2000: 120-122.

[Boehm 1998]

Barry Boehm & Alexander Egyed. “WinWin Requirements Negotiation Processes: A Multi-Project Analysis.” *Proceedings of the 5th International Conference on Software Processes*, 1998.

[Boehm 1976]

Barry Boehm. “Quantitative Evaluation of Software Quality,” 592 –605. *Proceedings of the 2nd International Conference on Software Engineering*. San Francisco, California, 1976.

[Bohannon 2005]

Mark Bohannon, “Science and Technology Conference for Chem-Bio Information Systems.” Joint Requirements Office for Chemical, Biological, Radiological, and Nuclear Defense (JRO for CBRND), October 2005.

[CBDP 2005]

Chemical and Biological Defense Program, Implementation Plan for Management of the Chemical and Biological Defense Program (including Supplemental Operating Procedures). Office of the Deputy Assistant to the Secretary of Defense for Chemical and Biological Defense (DATSD (CBD) 3050 Defense Pentagon, Room 3C257, Washington DC 20301-3050, August 2005.

[CJCSI 2007]

“Chairman of the Joint Chiefs of Staff, Joint Capabilities Integration and Development System, Instruction, CJCSI 3170.01F.” (1 May 2007).

[CJCSM 2007]

“Chairman of the Joint Chiefs of Staff, Joint Capabilities Integration and Development System, Manual, CJCSM 3170.01C.” (1 May 2007).

[CMMI 2007]

CMMI Product Team. *CMMI for Acquisition, Version 1.2*, CMU/SEI-2007-TR-017. Software Engineering Institute, Carnegie Mellon University, 2007.
<http://www.sei.cmu.edu/pub/documents/07.reports/07tr017.pdf>

[CMMI 2006]

CMMI Product Team. *CMMI for Development, Version 1.2*, CMU/SEI-2006-TR-008, ADA455858. Software Engineering Institute, Carnegie Mellon University, 2006.
<http://www.sei.cmu.edu/pub/documents/06.reports/06tr008.pdf>

[Coulter 1998]

Neal Coulter, Ira Monarch, & Suresh Konda. “Software Engineering as Seen through Its Research Literature: A Study in Co-Word Analysis.” *The Journal of the American Society for Information Science and Technology* 49, 13, November 1998: 1206-1224.
<http://www.asis.org/Publications/JASIS/v48n0597.html>

[Dixon 2005]

Nancy Dixon, Nate Allen, Tony Burgess, Pete Kilner, & Steve Schweitzer, *CompanyCommand: Unleashing the Power of the Army Profession*, Center for the Advancement of Leader Development & Organizational Learning (January 2005).

[DoA 2006]

Department of the Army, Headquarters, United States Army. Training and Doctrine Command, Fort Monroe, Virginia. TRADOC Pamphlet 525-3-3 Battle Command Concept Version 5-19, December 2006.

[DoA 2005]

Department of the Army, Headquarters, United States Army. Training and Doctrine Command, Fort Monroe, Virginia. Regulation 10-5, December 2005.
http://www.tradoc.army.mil/tpubs/regs/r10-5.htm#Para_2_4

[DoA 2003]

Department of the Army, Headquarters, United States Army. "Army Acquisition Policy, Army Regulation 70-1," Washington, DC, December 2003.

[Dunaway 1998]

Donna K. Dunaway, Dennis R. Goldenson, Ira Monarch, & David M. White. "How Well is CBA IPI Working? User Feedback." *Software Engineering Process Group Conference*. Chicago, Illinois, March 1998.

[Ellison 2004]

Robert J. Ellison, Andrew P. Moore, Len Bass, Mark Klein, & Felix Bachmann. *Security and Survivability Reasoning Frameworks and Architectural Design Tactics*, CMU/SEI-2004-TN-022, ADA443487. Software Engineering Institute, Carnegie Mellon University, 2004.
<http://www.sei.cmu.edu/pub/documents/04.reports/pdf/04tn022.pdf>

[Elm 2004]

Joseph P. Elm, Dennis R. Goldenson, Khaled El Emam, Nicole Donatelli, & Angelica Neisa. *A Survey of Systems Engineering Effectiveness--Initial Results*, CMU/SEI-2007-SR-014. Software Engineering Institute, Carnegie Mellon University, 2007.
<http://www.sei.cmu.edu/pub/documents/07.reports/07sr014.pdf>

[Firesmith 2006]

Donald Firesmith. *QUASAR: A Method for the Quality Assessment of Software-Intensive System Architectures*, CMU/SEI-2006-HB-001, ADA455116. Software Engineering Institute, Carnegie Mellon University, 2006. <http://www.sei.cmu.edu/pub/documents/06.reports/06hb001.pdf>

[Galt 2008]

Kimberly A. Galt. "Media Review: SPSS Text Analysis for Surveys 2.1." *SPSS Inc. Journal of Mixed Methods Research* 2, 2008: 284.

[Gangemi 2003]

A. Gangemi & P. Mika. "Understanding the Semantic Web through Descriptions and Situations," *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2003)*. Catania, Italy, November 2003. <http://www.loa-cnr.it/Papers/ODBASE-CONTEXT.pdf>.

[Goethert 2007]

Goethert, Wolf & Goldenson, Dennis. "Implementing CMMI[®] Measurement & Analysis Using Goal-Driven Measurement," European SEPG Conference, 2007.

[Goldenson 2008]

Dennis Goldenson & Ira Monarch. "Tailoring and Combining the CMMI-ACQ and Quality Models to Improve the Military's Requirements Process." 2008. <http://www.sei.cmu.edu/sepg>

[Goldenson 2007]

Dennis R. Goldenson. "Understanding CMMI Measurement Capabilities & Impact on Performance: Results from the 2007 SEI State of the Measurement Practice Survey." November 2007. <http://www.sei.cmu.edu/sema/presentations/measurement-survey2007.pdf>

[Greene 2003]

Harold J. Greene. *The Army Material Requirements Generation Process: A Process in Need of Change*. USAWC Strategy Research Project, U. S. Army War College, 2003. <http://stinet.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=ADA415080>.

[Guarino 1999]

Nicola Guarino, Claudio Masolo, & Guido Vetere. "OntoSeek: Content-Based Access to the Web." *IEEE Intelligent Systems*, May/June, 1999: 70-80.

[Guarino 2005]

Nicola Guarino & Mark A. Musen. "Applied Ontology: Focusing on Content." *Applied Ontology* 1, 1, 2005: 1-5.

[ISO/IEC 2001]

ISO/IEC "Software Product Quality - Part 1: Quality Model," June 2001. <http://engineers.ihs.com/document/abstract/LVYYPAAAAAAAAAAAAA>

[J7JE 2006]

"J7 Joint Experimentation, Transformation, and Concepts Division (JETCD) 2006 Assessment of JCAs Usefulness Data Call (Planner Level)," January 2006.

[Jackson 1995]

Michael J. Jackson. *Software Requirements & Specifications: A Lexicon of Practice, Principles, and Prejudices*. Reading, Mass.: Addison-Wesley, 1995 (ISBN: 0201877120).

[John 2004]

Bonnie John, Len Bass, Maria-Isabel Sanchez-Segura, & R. Adams, R.. “Bringing Usability Concerns to the Design of Software Architecture.” Proceedings of EHCI-DSVIS’04: The 9th IFIP Working Conference on Engineering for Human-Computer Interaction, July 2004.

[Krippendorff 2006]

Klaus Krippendorff. *The Semantic Turn: A New Foundation for Design*. Taylor & Francis: New York, 2006.

[Krippendorff 2004]

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Thousand Oaks, CA: Sage Publications, 2004 (ISBN: 0761915451).

[Krippendorff 1980]

Klaus Krippendorff. *Content Analysis*. Beverly Hills, CA: Sage Publications, 1980.

[Lockhart 2005]

Christopher E. Lockhart. “Soldier as a System.” Potential ACAT 1, Prepared for Milestone A Decision 7, February 2005.

[Mashiko 1997]

Mashiko, Y. & Basili, V. "Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement," 17-32 *Journal of Systems and Software*, Volume 36, 1. January 1997.

[Masolo 2005]

Claudio Masolo & Stefano Borgo. “Qualities in Formal Ontology,” 2–16. *In Foundational Aspects of Ontologies (FOnt 2005) Workshop*. Koblenz, Germany, 2005.

[Masolo 2003]

Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, & Alessandro Oltramari. “WonderWeb Deliverable D18, Ontology Library.”
<http://www.loa-cnr.it/Papers/D18.pdf>

[Mayk 2006]

Israel Mayk & Azad M. Madni. “The Role of Ontology in System-of-Systems Acquisition.” 2006 *CCRTS The State of the Art and the State of the Practice*. Santa Monica, CA, June 2006.
http://www.dodccrp.org/events/2006_CCRTS/html/papers/216.pdf

[McMaster 2003]

H. R. McMaster. *Crack in the Foundation: Defense Transformation and the Underlying Assumption of Dominant Knowledge in Future War*, S03-03. Center for Strategic Leadership, 2003.
http://www.au.af.mil/au/awc/awcgate/army-usawc/mcmaster_foundation.pdf

[Meyers 2006]

Craig Meyers, James D. Smith, Peter Capell, & Patrick Place. *Requirements Management in a System-of-Systems Context: A Workshop*, CMU/SEI-2006-TN-015, ADA449727. Software Engi-

neering Institute, Carnegie Mellon University, 2006.
<http://www.sei.cmu.edu/pub/documents/06.reports/06tn015.pdf>

[Mika 2003]

Peter Mika, Victor Iosif, York Sure, & Hans Akkermans, *Handbook on Ontologies*. New York: Springer, 2004 (ISBN: 3540408347), pages 455-476. Springer, 2003. http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/2003_ontohandbook_ens.pdf

[Monarch 2008]

Ira Monarch, Dennis Goldenson, Peter Capell, and Jim Wessel. "Tailoring and Combining the CMMI-ACQ and Quality Models to Improve the Military's Requirements Process." SEPG North America. Tampa, Florida, March 2008. <http://www.sei.cmu.edu/sema/presentations/monarch-acq.pdf>

[Monarch 2005]

Ira A. Monarch & Dennis R. Goldenson. "Categorizing Measurement & Analysis Needs in Software & Systems Engineering." *9th Annual Practical Software and Systems Measurement Users' Group Conference*. Keystone, Colorado, July 2005.

[Monarch 1995]

Ira A. Monarch. *An Experiment in Software Development Risk Information Analysis*, CMU/SEI-95-TR-014, ADA302320. Software Engineering Institute, Carnegie Mellon University, 1995. <http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr014.95.pdf>

[Morris 2004]

Edwin J. Morris, Linda Levine, Craig Meyers, Patrick Place, & Dan Plakosh. *Systems of Systems Interoperability*, CMU/SEI-2004-TR-004, ADA455619. Software Engineering Institute, Carnegie Mellon University, 2004. <http://www.sei.cmu.edu/pub/documents/04.reports/pdf/04tr004.pdf>

[Myers 1998]

Brad A. Myers. "A Brief History of Human Computer Interaction Technology." *ACM Interactions* 5, 2, 1998: 44-54.

[NDIA 2006]

National Defense Industrial Association (NDIA), Systems Engineering Division. "Task Group Report, Top Software Engineering Issues Within Department of Defense and Defense Industry," August 2006.

[Neuendorf 2001]

Kimberly A. Neuendorf. *The Content Analysis Guidebook*. Thousand Oaks, Calif.: Sage Publications, (ISBN: 0761919775). <http://www.loc.gov/catdir/enhancements/fy0656/2001004115-d.html>

[Nielsen 1994]

Jakob Nielsen. *Usability Engineering*. Boston: Academic Press, 1993 (ISBN: 0125184050).

[O'Brien 2005]

Liam O'Brien, Len Bass, & Paulo Merson. *Quality Attributes and Service-Oriented Architectures*, CMU/SEI-2005-TN-014, ADA441830. Software Engineering Institute, Carnegie Mellon University, 2005. <http://www.sei.cmu.edu/pub/documents/05.reports/pdf/05tn014.pdf>

[Ozkaya 2008]

Ipek Ozkaya, Len Bass, Raghvinder S. Sangwan, & Robert L. Nord. "Making Practical Use of Quality Attribute Information." *IEEE Software* 25, 2, 2008.

[Shneiderman 2004]

Ben Shneiderman & Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Boston: Pearson/Addison Wesley, 2004 (ISBN: 0321197860).

[Smith 2006]

A. E. Smith & M. S. Humphreys. "Evaluation of Unsupervised Semantic Mapping of Natural Language with Leximancer Concept Mapping." *Behavior Research Methods* 38, 2, 2006: 262-279.

[Standish 2004]

The Standish Group. "Chaos Research Results." Third Quarter Research Report, 2004.

[Standish 1995]

The Standish Group. "The CHAOS Report," 1995.
http://www.pm2go.com/sample_research/chaos_1994_1.asp

[Stone 1966]

Philip J. Stone, John Kirsch, & Cambridge Computer Associates. *The General Inquirer; A Computer Approach to Content Analysis*. Cambridge, Mass.: MIT. Press, 1996.

[Tolk 2005]

Andreas Tolk & Curtis Blais. "Taxonomies, Ontologies, and Battle Management Languages—Recommendations for the Coalition BML Study Group." 2005.

[Tufte 1997]

Edward R. Tufte. *Visual & Statistical Thinking: Displays of Evidence for Decision Making*. Cheshire, CT: Graphics Press, 1997 (ISBN 0961392134).

[Tufte 1983]

Edward R. Tufte. *The Visual Display of Quantitative Information*. Cheshire, Conn.: Graphics Press, 2001 (ISBN: 0961392142).

[USGAO 2008]

United States Government Accountability Office. *Defense Acquisitions: Assessments of Selected Weapon Programs* (GAO-08-467SP). March 2008. <http://www.gao.gov/new.items/d08467sp.pdf>

[USGAO 2004]

United States Government Accountability Office. *Defense Acquisitions: Stronger Management Practices Are Needed to Improve DoD's Software-Intensive Weapon Acquisitions*, GAO-04-393, March 2004. <http://www.gao.gov/new.items/d04393.pdf>

[van Genuchten 2001]

Michiel van Genuchten, Cor van Dijk, Henk Scholten, & Doug Vogel. "Using Group Support Systems for Software Inspections." *IEEE Software* 18, 3, May/June 2001: 60-65.
<http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/mags/so/&toc=comp/mags/so/2001/03/s3toc.xml&DOI=10.1109/52.922727>

[Walker 2007]

Ellen Walker. "Tech Views—Challenges Dominate Our Future." *DoD Software Tech News* 10, 3, October 2007.

[Wallace 2005]

William S. Wallace. "Network Enabled Battle Command," Royal United Services Institute, Defence Systems, 2005. <http://www.rusi.org/downloads/assets/Wallace.pdf>.

[Weber 1990]

Weber, Robert Philip. *Basic Content Analysis*. Thousand Oaks, CA: SAGE, 1990

[Woods 1997]

W. A. Woods. *Conceptual Indexing: A Better Way to Organize Knowledge*, SMLI TR-97-61. Sun Microsystems Laboratories, Mountain View, CA, 1997.
<http://www.sun.com/research/techrep/1997/abstract-61.html>

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE September 2008		3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Requirements and Their Impact Downstream: Improving Causal Analysis Processes Through Measurement and Analysis of Textual Information			5. FUNDING NUMBERS FA8721-05-C-0003	
6. AUTHOR(S) Ira A. Monarch, Dennis R. Goldenson, & Lawrence T. Osiecki				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2008-TR-018	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-2008-018	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) Requirements documents, test procedures, and problem and change reports from a U. S. Army Software Engineering Center (SEC) were analyzed to identify, clarify, and begin categorizing recurring patterns of issues raised throughout the product life cycle. Semi-automated content analysis was used to identify underlying patterns in the SEC documents. Automated tools and techniques were used to support efficient search and related semantic analysis that would not be possible manually. Discussions with Army personnel were used to confirm and elaborate initial findings and interpretations. The same analytic methods can be used as a basis for novel, proactive causal analysis processes. One of the patterns identified here suggests that usability is not sufficiently articulated and quantified early in the product life cycle. While the SEC has established exemplary processes to handle usability-related issues when they arise, some of them might be mitigated or prevented by documented consideration upstream.				
14. SUBJECT TERMS Requirements engineering, traceability, validation, maintenance, testing, usability, quality attributes, process and quality modeling, CMMI, measurement and analysis, causal analysis and resolution, content analysis, automated text analysis, semantic analysis, ontology			15. NUMBER OF PAGES 75	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	